

הכנס הארצי לעיבוד נתונים

חיפה תשל"ז 1976

קובץ ההרצאות



איגוד ישראלי לעיבוד אינפורמציה

הכנס הארצי לעיבוד נתונים

חיפה תשל"ז 1976

קובץ ההרצאות



איגוד ישראלי לעיבוד אינפורמציה

מספר מאמרים הוגשו בכנס אך נוסחם המלא לא הועבר למערכת,
מאמרים אלה אשר לא נכללו בקובץ, מפורטים להלן:

קבוצת דיון:	שם המחבר:	שם המאמר:
מס' 3	א. רייטר	"SOME EXPERIMENTS IN DIRECTORY ORGANIZATION — A SIMULATION STUDY"
מס' 3	י. רגב	COMPUTER SELECTION BENCH-MARKING AND TIME SHARING
מס' 8	א. סגל	DYNAMIC ROUTINE IN COMPUTER COMMUNICATION NETWORK
מס' 8	ד. לוין	ביזור מאגרי נתונים ברשתות מחשבים, בעיות ופתרונות
מס' 10	א. שץ	מחשבים ותקשורת - טכנולוגיה ויישומים
מס' 11	מ. נאמן	שחזור והתגברות על תקלות במערכת מאגרי נתונים
מס' 11	נ. למלבאום	הסבת מערכת תכניות במשרד התקשורת
מס' 12	א. פרנקל	בחירה, תכנון ומדידת ביצוע של מערכות, אחסון ואחזור מידע
מס' 12	ו. מוסקוביץ	LINGUISTIC TECHNIQUES FOR AUTOMATIC DOCUMENT RETRIEVAL AND FACT RETRIEVAL SYSTEMS

המאמרים בקובץ זה הובאו לדפוס

בצורה שהוגשו ע"י מחבריהם ועל אחריותם

תוכן העניינים

עמוד

עמוד

קבוצת דיון מס׳ 7

קבוצת דיון מס׳ 8

קבוצת דיון מס' 1:

"מערכות מידע ניהולי".

יו"ר: פרופ' נ. נוימן

מ. שיזף

א. שייביץ

פ. עין דור

ע. שגב

מערכת "בסיס נתונים זעיר"

מיכאל שיזף - משרד הבטחון

תקציר

מתוארת מערכת מידע נהולי המבוססת על "בסיס נתונים זעיר". המערכת מיועדת למעקב תקציבי בפרויקט הנדסי, בבסיס הנתונים נרשמות כל הפעולות הכספיות והמערכת עונה על שאילתות הנוגעות למצב התקציבי והעדכני, זאת במספר חתכים. למערכת שני מאפיינים יוצאי דופן שבגללם עשוי להיות בה ענין כללי :-

* המערכת פועלת על "מחשב שולחני" - DESK CALCULATOR
* המערכת הוקמה ע"י המשתמש ולא ע"י אנשי ענ"א מקצועיים.

מבוא

מקובל שמדענים ומהנדסים כותבים לעצמם את התכניות המשמשות אותם בעבודתם. פחות מקובל שתכניות נהול וכספים תכתבנה ע"י המשתמשים. אולי טבעי, לכן, שדווקא יחידה הנדסית תכין לעצמה תכנית נהול כספית. הנהלת החשבונות של הפרויקט מתנהלת במסגרת המערכת המרכזית של המשרד. מערכת זו אינה עונה על הצרכים המיוחדים של היחידה: אי אפשר לקבל מידע בחתכים המענינים אותנו וההתקשרות הנה מסורבלת. פתרנו את הבעיה ע"י כך שתכננו לעצמנו "מערכת מידע נהולי" הפועלת על מחשב שולחני הקיים ביחידה.

המחשב השולחני שברשותנו ניתן לתכנות בשפת תכנות אלגברית. יש לו זכרון פנימי של כ - 7K בתים, ומחוברת אליו קסטה המהווה זכרון נוסף של כ - 30K בתים. התכנות במכונה זו הינו תהליך אינטראקטיבי המתבצע ON - LINE כשהמתכנת עובד עם המכונה. מערכת ההפעלה (הבנויה בחומרה) כוללת תכנית פרוש- INTERPRETER - המאפשרת עריכה נוחה מאוד, כולל בצוע תקוני שגיאות תוך כדי הרצת התכנית.

בסיס הנתונים מסוגל לקלוט כ - 1500 פעולות, היקף התואם שנת עבודה. ניתן לפנות למערכת בשאילתות על המצב התקציבי העדכני, זאת במספר חתכים ובהירכיה בת ארבע רמות. אפשר גם לקרוא לכל פעולה שנעשתה בעבר, הקריאה יכולה להעשות לפי מספר מאפיינים. הצגת שאילתות למערכת, וגם הזנת הנתונים לתוכה אינה מחייבת ידיעה מראש של פורמט מוגדר, המערכת מנחה את המשתמש בה ע"י הצגת שאלות על גבי הצג.

2./..

בדיעבד נראה שעל ידי הקמת המערכת בצורה זו השגנו מספר יתרונות :-

* הבנה מלאה של המערכת והפוטנציאל הגלום בה.

* נןחיות רבה בהכנסת שנויים ותוספות.

* גישה בלתי אמצעית המאפשרת עדכון שוטף וקבלת חשובות מידיות לשאילתות.

נראה לי שכדאי להביא את ספור המערכת שלנו לידיעת הכלל משני טעמים :

* לעורר מודעות לשמושים נוספים אצל בעלי ציוד דומה לשלנו, אשר בדרך כלל מנוצל לשמושים טכניים ומדעיים בלבד.

* לעורר ענין אצל אנשי יחידות המחשב המרכזיות של גופים גדולים במתן אפשרות ליחידות לבנות לעצמן מערכות מידע נהולי ובסיסי נתונים על המחשב המרכזי. ברור שבאמצעות צג המחובר למחשב גדול, המצויד במערכות תוכנה מתאימות, היינו יכולים לבנות לנו מערכת בעלת כל היתרונות שפורטו לעיל ובלי המגבלות שהוכתבו ע"י ציוד קטן ואיטי.

"נתוח המערכת"

מדובר ביחידה העוסקת בנהול פרויקט הנדסי. ביחידה אנו זקוקים למידע נהולי עדכני על הצד התקציבי של בצוע התכניות. המידע שאנו זקוקים לו הוא מהסוג של : מה ההיקף העדכני של ההתחייבויות ? מהן יתרות התקציב ? מה היה היקף המימון בתקופה מסוימת ? (פרוט מלא, ראה בטבלה מס' 1). השאלות הנ"ל נשאלות ביחס לתקציב הכולל או לגבי פריט מיוחד באחת מהרמות.

המערך התקציבי שאנו עוסקים בו הינו הירכיה בת ארבע רמות :

* רמת הפעילויות.

* רמת הסעיפים.

* רמת קבוצות הסעיפים.

* רמת התקציב הכולל.

שייכות הפעולות לסעיפים הנה חד משמעית; כל פעילות שייכת לסעיף מסוים. ענין קבוצות הסעיפים שונה. קיימות מספר חלוקות לקבוצות סעיפים, כאשר כל חלוקה מהווה מיון של הסעיפים לפי חתך מסויים.

העלויות של הנושא בו היחידה עוסקת מתנהלות, כמובן, במערכת הנהלת החשבונות של המשרד. אולם, לא יכולנו להעזר במערכת זו לצרכים שלנו. אי אפשר היה לקבל נתונים בחתכים המעניינים אותנו, ובשלב הנוכחי היא אינה בנויה לעבודה בשיטה של תשובה על שאילתות.

3./..

* צג אלפא - נומרי.

* מדפסת אלפא - נומרית. אורך השורה הוא 16 סימנים. פורמט מצומצם זה הינו מחסרונותיה של המכונה בשמוש בו אנו דנים.

* אמצעי קלט - פלט לכרטיסים מגנטיים המשמשים לאחסון תכניות ונתונים.

* זכרון פנימי של כ - 1400 "רגיסטרים". זכרון זה משמש לקליטת תכניות ונתונים גם יחד. ב"רגיסטר" זה ניתן לרשום מספר עשרוני של 12 ספרות סיגניפיקנטיות, 2 ספרות חזקה וסימן, כמובן. מכאן שהזכרון הפנימי הינו שווה ערך לזכרון בינרי של כ - 7K בתים. כאשר הזכרון משמש לקליטת תכניות הוא שווה ערך ל-16 K בתים.

* למחשב מחובר רשם סרט המהווה זכרון נוסף. רשם הסרט קולט קססות שהתכולה שלהן כ - 6 K "רגיסטרים" שהם שווי ערך לכ - 30 K בתים. בקסטה ניתן לאחסן תכניות וגם נתונים. את הפקודות להפעלת הקססה אפשר לשלב בתכניות המחשב.

* תכנות המחשב הינו תהליך אינטרקטיבי המתבצע ON - LINE. לדעתי זוהי תכונתה המעניינת ביותר של המכונה מבחינת המשתמש. מערכת ההפעלה כוללת תכנית פרוש המאפשרת עריכה נוחה מאוד, (קיימות פקודות DELETE, INSERT, FORWARD, BACK). המכונה מציינת חלק מהטעויות תוך כדי כתיבת התכניות. בנוסף לכך קיימת אפשרות לתקן שגיאות תוך הרצת התכנית. כל זה מקל מאוד את התכנות, וניתן לומר ש"מחבב" אותו על ה"מתכנת" שהתכנות עבורו הוא רק כלי ולא העיקר.

התכנה

המערכת משמשת לאגירת המידע התקציבי ולמתן תשובות מעודכנות על סמך מידע זה. הזנת הנתונים נעשית האמצעות מה שאנו מכנים "פעולות" והמידע מתקבל ע"י שאילת "שאילתות". ה"פעולות" וה"שאילתות" מתבצעות ע"י המשתמש בפעולה אינטרקטיבית עם המחשב.

5./..

הציוד

ברשות היחידה "מחשב שולחני" - DESK CALCULATOR
המשמש לעבודות הנדסיות. מדובר במכונה דגם 9820 A של חברת HEWLETT PACKARD.
תכונותיה העיקריות של המכונה הן (בקונפיגורציה שברשותנו) :-

* שפת תכנות אלגברית, הכוללת את הפקודות הבאות:
  - פקודות מיעון JUMP, GO TO, GO TO SUB, RETURN
  - פקודת תנאי (IF) שהארגומנט שלה יכול להיות מורכב למדי.
  - פעולות ופונקציות מתמטיות משוכללות.

  בציור מס' 1 מוצגת דוגמא של קטע תכנית כפי שהיא מודפסת ע"י המכונה.

ציור 1 :- שגרת "דיאלוג" כדוגמא לתכנית כפי שהיא מודפסת ע"י המחשב.

4./..

שאילתות	פעולות
1. נתוני פעילות מעודכנים .	1. קביעת תקציב בסעיף.
2. נתוני סעיף מעודכנים.	2. הגדלת תקציב בסעיף.
3. סכום נתוני קבוצת סעיפים.	3. הקטנת התקציב בסעיף.
4. כנ"ל + פרוט נתוני הסעיפים.	4. העברה מסעיף לסעיף.
5. סכום נתוני התקציב הכולל.	5. יצירת התחייבות בסעיף.
6. כנ"ל + פרוט קבוצות הסעיפים.	6. מימון בסעיף.
7. כנ"ל + פרוט לסעיפים.	7. מימון בסעיף ע"ח התחייבות.
8-12. כמו 3 עד 7 אולם רק נתוני תקציב.	8. פתיחת פעילות חדשה.
13. כל הפעולות בתקופה מסויימת.	9. מימון בפעילות.
14. כל הפעולות בסעיף מסוים.	
15. כל הפעולות בפעילות מסויימת.	

טבלה 1 :- רשימת הפעולות והשאילתות שהמערכת מבצעת.

6./..

בסיס הנתונים בנוי ב-3 רמות :

* נתוני הסעיפים.

* נתוני הפעילויות.

* רשום כל הפעולות - ה"יומן".

שאלתות ברמות אלו יש להן שגרות הפונות הישר לבסיס הנתונים. לשאלתות ברמות הגבוהות יותר שגרות הכוללות בתוכן את המידע ביחס לסעיפים אליהן מתיחסת השאלתה. זוהי דוגמא ברורה של השפעת מגבלות הזכרון, היה הכרח לכלול פריטי מידע בתכנית תוך פגיעה בגמישות המערכת.

התכנית כוללת 5 פרקים ראשיים (ראה ציור 2):

* שגרת הפעלה.

זהו BOOT STRAP הנמצא על כרטיס מגנטי. בהפעלתה היא מכניסה את בסיס הנתונים ואת התכנית מהקסטה לזכרון הפנימי.

* "הדיאלוג".

בשגרה זו המשתמש מזין את הנתונים או שואל את השאלתות.

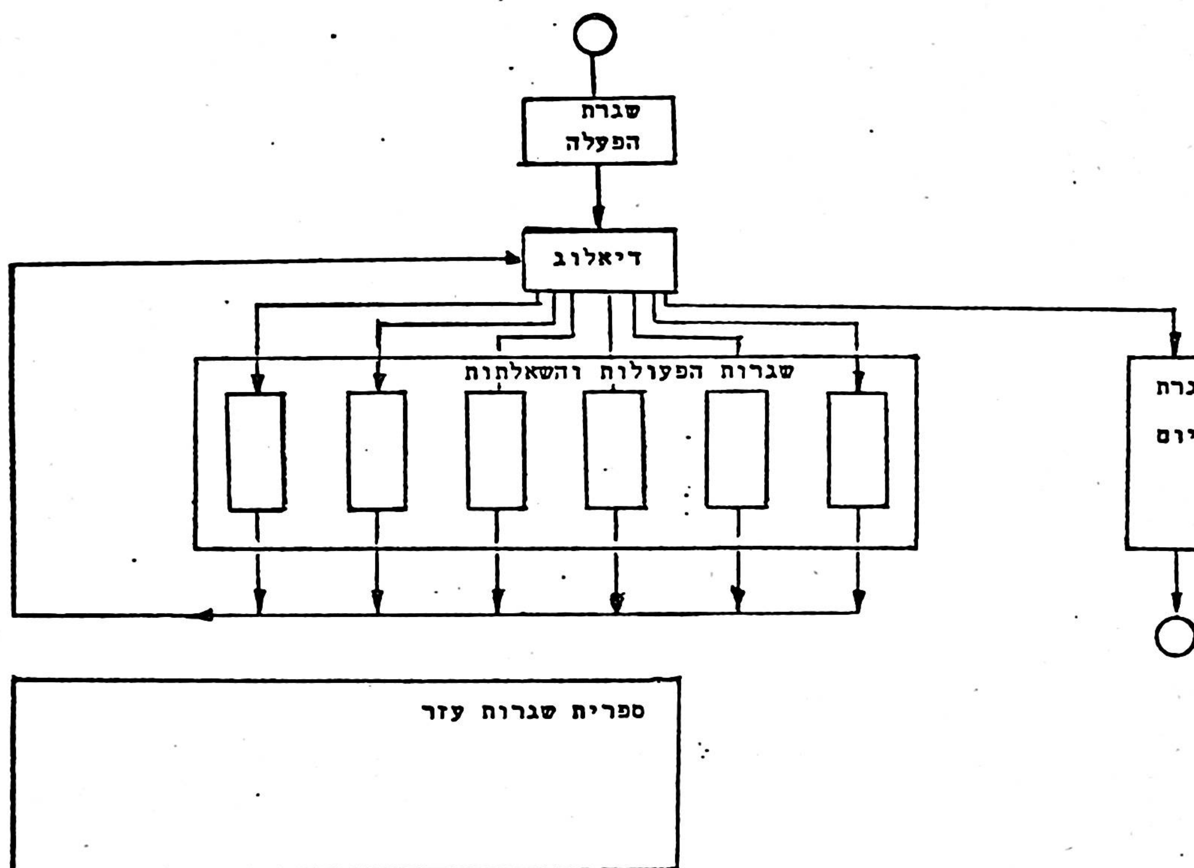
* שגרות הפעולות והשאלתות.

שגרות אלו מבצעות את רשום הנתונים או את הוצאת הנתונים והצגתם.

* ספרית שגרות העזר.

* שגרת סיום.

שגרה זו מטפלת בהחזרה לקסטה של בסיס הנתונים המעודכן.



ציור 2 : מבנה התכנית.

תפעול המערכת הוא פשוט מאוד היות והמכונה מנחה את המשתמש על ידי הצגת שאלות על גבי הצג. להלן דוגמת דיאלוג בין המכונה למשתמש.

המשתמש מפעיל את המחשב והקסטה, מכניס את כרטיס ההפעלה ולוחץ "הרץ תכנית". התכנית והנתונים מעברים מהקסטה למחשב,התכנית מתחילה לרוץ ומתחיל הדיאלוג כאשר המחשב שואל על גבי הצג.

המחשב: "התאריך?"
המשתמש עונה באמצעות לוח המקשים.
המחשב: "סוג פעולה?"
המשתמש: רושם את מספר הפעולה.
המחשב: "הסכום?"

הדיאלוג נמשך עד שהמחשב מקבל את כל הנתונים הקשורים בסוג הפעולה שבה עוסקים. בתום הדיאלוג ולפני הזנת הנתונים לבסיס המידע, מתבצעת פעולת אמות משולבת בין המחשב למשתמש. כל הנתונים מדפסים (ראה דוגמה בציור 3) והמחשב שואל: "שגיאה?" ובמקרה של טעות אפשר לתקן.

ציור 3: קטע מה"דיאלוג" בו המחשב מדפיס את הנתונים לבדיקת המשתמש.

מגבלות

לאחר הפעלת המערכת החלו להתגבש דרישות למידע נוסף אם להעמקתו ואם להרחבתו. דרישות שנראו בעבר כמיותרות או מרחיקות לכת מדי. אולם עכשיו, לאחר שהתרגלנו כבר לרמת מידע החדשה, הרי שהמידע הנוסף נראה כבר כהכרחי. חלק מדרישות אלו ניתן יהיה לממש. רובן הן, כנראה, מעבר לניתן לבצוע על "מחשב שולחני" שמטבעו זכרונו מוגבל ופעולתו איטית. להלן שתי דוגמאות של דרישות שכנראה לא נוכל לממשן:

* בסיס מידע רב שנתי ~~שאפשר~~
  - סכומי נתונים רב שנתיים.
  - צבירת יתרת התחיבויות משנים קודמות.
  - רשום התחיבויות לשנים הבאות.

* טבלת תרגום שתאפשר גישה לסעיף לפי מספרי כנוי שונים שיש לו אצל גופים שונים.

9./..

סכום

מערכת המידע הנהולי שהקמנו עונה על הדרישות כפי שהגדרנו אותן בתחילת העבודה ומשרתת אותנו לשביעות רצוננו. אולם כנראה שהנה המירב שאפשר לעשות על מחשב שולחני.

מערכת אינטראקטיבית לניהול ספריות צה"ל

סרן שייביץ אביתר

O.L.I.S - ON LIBRARY INFORMATION SYSTEM

1. מבוא

במאמר זה מובא תאור מתומצת של מערכת אינטראקטיבית לניהול ספריות.
המערכת נבנתה כיחידה שלמה על מחשב זעיר ובכך ייחודה המשמעותי.
באמצעות מערכת זו ניתן לבצע את כל הפעילויות הנובעות מעבודתה של ספריה והכוללת - אחזורי מידע עפ"י מילות-מפתח, הוספת ספרים חדשים לספריה, עדכון ישנים, ניהול רשימות תפוצה וכדי'.

1.1 מטרת המערכת

קליטה: רשום ממוכן של כל פרטי האינפורמציה הדרושים לגבי הספריה תוך פקוח על שלמות ונכונות האינפורמציה המוזנת.

אחזור: אחזור אינפורמציה יעיל ופשוט המבוסס על מילות-מפתח שהוצאו לספרים שנרשמו. תנאי אחזור בוליאניים מסובכים.

הצגה: הצגת אינפורמציה מירבית למשתמש לאחר כל שלב אחזור כדי לאפשר לו להגיע לקביעה נכונה בדבר נחיצות הספר המאוחזר לעינינו.

תפוצה: ניהול רשימות תפוצה בחתכים של מחזיקי ספרים וספרים.

דו-שיח: ביצוע כל הפעולות המוזכרות ע"ג מסך המחשב בזמן תגובה מיידי בגישה אינטראקטיבית מלאה.

1.2 האמצעים

מחשב זעיר מסוג NOVA מתוצרת חברת DATA-GENERAL בנפח זכרון של 32KW
מהירות זכרון 1000N/S. שפות תכנות קיימות: BASIC, FORTRAN, ASSMBLER, ALGOL. מערכת הפעלה RDOS/4 התומכת בדיסקים.
שני דיסקים בעלי נפח כולל של 10/BM בעלי מהירות העברה של 195,250 B/S
צג אלפא נומרי 2480 ch.
מדפסת בעלת קצב הדפסה ממוצע של 600L/S,
קורא כרטיסים מעל מהירות ממוצעת של 300C/M.

1.3 מגבלות המחשב

מגבלת זכרון - נפח כולל של 32KW כאשר מערכת ההפעלה מנצלת 11KW מהנ"ל.

מגבלת דיסקים - נפח הדיסקים מוגבל ומצריך דחיסת נתונים.

תוכנה - לא קיימת תוכנה סטנדרטית המבצעת את האמור ויש צורך בפתוח עצמי.

1.4 דרישות מהמערכת

א. זמן תגובה קצר.

ב. כמות גדולה של מילות-מפתח, ספרים, מסמכים.

ג. גישה אינטראקטיבית מלאה.

ד. שליפה סידרתית והדרגתית תוך זכרון הפעולות הקודמות.

ה. מתן אפשרות לתנאי שליפה מסובכים.

ו. מילות-מפתח בעלות אורך בלתי קבוע.

ז. הגדרת מילים נרדפות למילות המפתח.

ח. גמישות לשינויים.

ט. מעברים פשוטים משלב השליפה לשלב הקליטה ולהיפך.

י. קבלת העתק מתהליכי השליפה, ע"ג נייר מודפס.

2. הפעולות האפשריות במערכת

2.1 פעולות הקליטה

אוסף הפעולות הנ"ל נועד לשמש כלי עזר בידי הספרן בכל הקשור בפעילויות קליטת ספר חדש לספריה, הזנת המידע המיועד בבסיס הנתונים, שינויים במידע הקיים לגבי ספרים ישנים, ניהול רשימות תפוצה והכנסת מילות מפתח לספר.
להלן רשימת הפעולות הקיימת:
קליטת ספר חדש (מובחה אוטומטית), שיבוי מידע ישן, הוספה/גריעה של מילות מפתח לספר, הוספה/גריעה של רשימת תפוצה לספר, הקצאה/ביטול של מילים ברדפות למילת-מפתח, הצגת תקציר הספר ע"ג המסך, הדפסת תקציר הספר ע"ג המדפסת.

1.2 פעולות האחזור

האופרטורים הבוליאניים הקיימים בפעילויות אלו -

* = AND
\+ = OR
! = AND NOT

ביתן להגדיר בפעולות האחזור כל בטוי בוליאני (על מילות מפתח) בכל רמה רצויה של סוגריים. התשובה המתקבלת כתוצאה מהאחזור כוללת את מספרי הספרים השונים העונים לבטוי הבוליאני שהוגדר.
כל תשובה כזו מסופררת ונצברת כך שביתן לחזור לשלב קודם ולאחזר עפ"י מס' התשובה במקום הביטוי הבוליאני, להבהרת הנושא בעיין בדוגמאות אחזור אפשריות. (בדוגמא הסימן 'U' - מציין משתמש והסימן "C" - מציין את תגובת המערכת, הסימן "O.K" מציין בכוננות המערכת להכנסת תגובת המשתמש).

"C"	O.K	
"U"	MOTORS	(המשתמש מאחזר את הספרים העוסקים במנועים)
"C"	100 BOOKS ANSWERED REQUEST #1	
	0009,0501,...	(מספרי הספרים הנ"ל)
	O.K	
"U"	MOTORS * CARS	(המשתמש חפץ רק במנועי מכוניות)

"C"	33 BOOKS ANSWERED REQUEST #2
009,1932,1939...
O.K
"U"	#2!FIAT	(המשתמש חפץ במנועים שאינם מתוצרת FIAT)
"C"	25BOOKS ANSWERED REQUEST #3
1931,1994...
O.K
"U"	$SHOW #3	(המשתמש מבקש להסתכל בתקצירי ספרים אלו
"C"	:	ע"פ סדר הופעתם בתשובה 3# כדי לברור
:	את הנחוצים לו).

בפעולות אחזור אלו ניתן גם להתייחס לרשומות תפוצה ולברר אלו ספרים מחזיק פלוני אלמוני.

לדוגמא:

"C"	O.K
"U"	ZVI AVIRAM
"C"	5 BOOKS ANSWERED REQUEST #4
0500,0609,0667,0891,0999
:
:

עיקרי השיטה - הגישה לאינפורמציה

3 שטח אחסנה קבוע

עקב הצורך בזמני תגובה מהירים ודחיסת נתונים מקסימלית התעורר הצורך בקביעת עקרון ראשוני חשוב של הכנסת פריטי המידע לרשומות בעלות אורך קבוע.
עקרון זה יאפשר פתוח שיטת גישה מהירה ונוחה המתבססת על אינדכס מספר הרשומה.
ברור שכדי לאפשר הכנסת מידע בעל אורך משתנה (כגון: מילות-מפתח שאורכן אינו קבוע) היה צורך לפתח מכניזם לשמירת עודף המידע שאינו נכנס לשטח קבוע. פותח מכניזם ליצירת ובנית שטחי גלישה (OVERFLOW-AREA) שגם הם בעלי אורך קבוע.
יתכן איפוא שפריט מידע מסוים יאוחסן בהמשכים בכמה שטחי אחסון כאשר הראשון הוא הבסיסי והשאר שטחי גלישה.

3 שיטת ארגון המידע ע"ג הדיסק

שיטת אחסנה זו תנאמת גישות ישירות לפריטי מידע והיא מבוססת על חלוקת שטחי אחסנת האינפורמציה לבלוק-אינדכס (INDEX-BLOCK) ולבלוק נתונים (DATA-BLOCK).
בלוק לצרכים אלו יוגדר כשטח אחסנה פיזי בו 512 בתים.
בבלוק האינדכס מצויות הכתובות היחסיות של בלוקי הנתונים השונים.
בשיטת ארגון זו מתאפשרת גישה ישירה לבלוק המידע עצמו באמצעות כתובתו המצויה באינדכס. בלוק האינדכס מכיל הצבעות ל-255 בלוקי-נתונים. לכן ניתן באמצעות מספר מועט יחסי של בלוקי האינדכס להצביע על בלוקי נתונים רבים. שיטת ארגון זו אינה קובעת שמקומם הפיזי של בלוקי הנתונים ע"ג הדיסק חייב להיות ברצף אחד, להיפך, היא מאפשרת פיזור שרירותי בהתאם למקומות הפנויים ע"ג הדיסק.
בשיטה זו אין גם הקצאה ראשונית של כל בלוקי הנתונים ע"ס מספר ההצבעות בבלוקי האינדכס אלא מתהווה הקצאה דינמית של בלוק נתונים בכל פעם ראשונה שהתבקשה גישה לבלוק מסוים.
כדי לאפשר גישה מהירה לבלוק נתונים מסוים יש צורך לדעת את המספר הסידורי של הבלוק המבוקש ואותו למסור כפרמטר למכניזם המטפל בהבאתו לזכרון. הסבר על אופן קביעת מספר הבלוק יובא בהמשך.

3 גישה באמצעות מילת-מפתח

כל עקרון האחזור המתואר במערכת זו מבוסס על מילות מפתח הנקבעות לכל ספר ואשר הן מהוות מפתח גישה לאחזור. לכן התעורר הצורך למצוא שיטה בסיסית לביצוע גישה באמצעות מילות מפתח לרשומת הספרים אותם מייצגת מילת המפתח האמורה.

3.4 גישה ישירה למילות המפתח

כדי לשמור על המוזכר בסעיף הקודם ויחד עם זאת לאפשר גישה ישירה נקבעו העקרונות:

א. כל הוספה של מילת מפתח חדשה למערכת תעבור את הפעילויות המוזכרות בסדר הבא: המילה תכנס לקובץ "שמות" בצורתה המקורית ותקבל מספר סידורי עולה.
המילה תעבור הסבה באמצעות פונקציה כמתואר ותרשם ברשומת הנתונים (המכילה את מספרי הספרים שמילת מפתח זו מייצגת). לרשומת נתונים זו ניתן יהיה לגשת ישירות.

ב. יש למצוא פונקציה המתארת באופן חד-ערכי, עד כמה שניתן את מילת-המפתח ולקבוע את ערך הפונקציה הנ"ל כמיצג בעל אורך קבוע של מילת-המפתח ברשומת נתונים אליה תעשה גישה ישירה. (ראה 3.1)

3.5 הפונקציה לייצוג מילות מפתח

כפי שתואר עלינו למצוא פונקצית ייצוג למילות המפתח.
לפונקציה המבוקשת מספר דרישות -

א. פשוטה - מהירה לחישוב כדי לא להוריד את זמן התגובה של המערכת.

ב. מייצגת- חד-ערכית עד כמה שאפשר.

ג. דחיה - דחית מילים בעלות ערך פונקציה שונה מזו הנבדקת. אך כאשר פונקציות תואמות אין הדבר מצביע עדיין על זהוי המילה. (כיון שאין חד ערכיות שלמה).

הפונקציה שנבחרה נחלקת לשלוש תת-פונקציות פשוטות -

3.5.1 פונקציה א: הוצאת אות מסוימת ממילת המפתח ושמירתה במבנה תוים בשטח בן בית אחד.

3.5.2 פונקציה ב: לגבי פונקציה זו מילת המפתח המקורית היא אוסף זוגות אותיות ע"פ סדר הופעתן במילה. מתבצע סכום הערך האוקטאלי של זוגות אלו, בהזזות ביטים. מסכום זה נלקחים 8 ביטים המשמעותיים ביותר.

3.5.3 פונקציה ג: פונקציה זו דומה לקודמתה אך כאן מתבצעות החלפות בין זוגות האותיות, בלקח הערך האוקטאלי של זוג אותיות והערך של המוחלף.

3.5.4 הפונקציה המיצגת: כל שלוש תת-הפונקציות שתוארו לעיל תורמות כל אחת את חלקן לפונקציה המיצגת הכוללת ואנו מקבלים פונקצית דחיה המורכבת משלושה חלקים ותופסת עבור כל מילת מפתח שטח זכרון קבוע בן 3 בתים בלבד.
מהירות חישובן של שלוש פונקציות אלו גבוהה ביותר מכיון שהן משתמשות בסט-הפקודות הקיימנת במחשב (פקודות החמרה) ומנצלות את יכולתן בצורה היעילה ביותר. בבדיקות נסיוניות על מדגם בן 1024 מילות מפתח לא נתגלו שתי מילים שהפונקציה המיצגת היתה זהה לשתיהן, למעלה מ-95% ממילות המפתח שונות כבר בתת-הפונקציה הראשונה ממילת מפתח נתונה.

3.5.5 אלגוריתם הדחיה: פעולת אלגוריתם הדחיה בחיפוש נעשית באופן הדרגתי. למעשה בעזרת אלגוריתם זה נזהה האם שתי פונקציות מיצגות של שתי מילות-מפתח זהות ביניהן. הדרגתיות האלגוריתם נקבעת בעזרת שלושה תת-חלקי הפונקציה כאשר בכל שלב מחושב החלק הנדון.
האלגוריתם:

א. אם שני ערכי הפונקציה (א) לא זהים?	-דחה-
ב. אם שני ערכי הפונקציה (ב) לא זהים?	-דחה-
ג. אם שני ערכי הפונקציה (ג) לא זהים?	-דחה-
ד. אם מילות המפתח עצמן לא זהות?	-דחה-
ה. במצאה המילה המבוקשת.	-קבל-

בעזרת אלגוריתם זה ניתן לבדוק האם הפונקציה המיצגת הקיימת ע"ג הקובץ זהה לפונקציה המיצגת אותה מאחזר המשתמש, דהיינו נאם נמצאה מילת המפתח המבוקשת.

בכל המילים שנבדקו התברר שפונקצית הדחיה ללא סעיף ד. דוחה כאשר מילת המפתח בדיסק לא זהה למילת המפתח המבוקשת ע"י המשתמש.

3.5 קביעת מספר בלוק-הנתונים

הבעיה המתעוררת כאן קשורה בשאלה באיזה בלוק נתונים יש להכניס רשומת מידע כזו למערכת, או באיזה מספר בלוק נתונים יש לחפש מילת מפתח הקיימת כבר? כדי להשתמש במכניזם (סעיף 3.2) לחיפוש בלוק נתונים בשיטת הארגון רנדום יש לתת למיכניזם את מספר הבלוק המבוקש, מכאן שיש למצוא דרך למציאת מספר בלוק כפונקציה של מילת המפתח. אז ניתן יהיה כתוצאה מהפעלת הפונקציה להגיע לבלוק בו אמורה להמצא רשומת המידע המייצגת את מילת המפתח. מציאת מספר הבלוק המבוקש מתבצעת האמצעות פונקצית HASH מיוחדת.

דרישות מפונקציות HASH

א. חובה להביא להתפלגות רנדומית של מפתחות החיפוש על פני כל איבטרבל הכתובות האפשרי. היא חייבת להקטין את הסיכויים שמפתחות כמעט זהים יקבלו כתובת יחסית זהה בקובץ הנתונים. זוהי למעשה הדרישה להגיע למינימום חזרות (COLLISIONS) זהות עם מפתחות שונים. תנאי זה מוגדר במסגרת DATA INDEPENDENT.

ב. כאשר מתקיימות חזרות כמתואר יש להבטיח מציאת פתרון כגון: כתובת אלטרנטיבית.

ג. היות וכל מילה המוכנסת לקובץ הנתונים וכל מילה המבוקשת לאחזור עוברת דרך פונקציה זו יש להבטיח שהפונקציה תהיה פשוטה נמהירה בחישוב MACHINE DEPENDENT.

ד. בזבוז מינימלי של מקום האחסנה בקובץ הנתונים.

4 פונקצית ה-HASH הנבחרת

4.1 שיטת העבודה

הפונקציה הנבחרת משתמשת באופן בסיסי בשיטת ארגון הקבצים המוזכרת RANDOM ומפזרת את מילות המפתח בין בלוקי הנתונים השונים. בכל בלוק כזה מקום ל-51 רשומות מידע המיצגות כמוזכר את מילות המפתח. מספר בלוקי הנתונים נקבע להיות 512 משיקולים שיוסברו בהמשך וזאת מתוך מטרה לאפשר מקום ל-20,000 מילות מפתח (ראה תנאי צפיפות). פונקצית ה-HASH מוצאת בשלב האחזור את מספר בלוק הנתונים בו צריכה מילת המפתח להמצא, ובשלב הקליטה את המקום בו תושם מילה חדשה.
הבלוק נקרא לזכרון ועליו נעשית סריקה סדרתית כדי למצוא בין רשומות הייצוג שיכולות להיות בו את הרשומה המבוקשת. כאשר נמצאת הרשומה מילת המפתח המבוקשת נמצאת במערכת, אחרת זוהי מילה חדשה.
בבלוקים אלטרנטיביים לבלוקים מלאים נקבעים הבלוקים שמספרם גבוה מ-512 באופן של כל 8 בלוקים מקוריים קיים בלוק אלטרנטיבי אחד.

4.2 הפונקציה

מתבצעת סריקה על מלות המפתח בזוגות אותיות. כאשר נלקח הערך האוקטלי של כל זוג אותיות כולל הזזה סיבית בערך זה. בזוג האותיות הראשון הזזה סיבית אחת ימינה ובזוג השני שמאלה וכך הלאה. מתקבל סכום אוקטלי בשדה בן 16 סיביות, משדה זה בלקחים 9 סיביות אמצעיות כך שמתקבל בסופו של תהליך מספר המקיים $0 \leq j \leq 511$. מספר זה יהווה את מספר "בלוק-הנתונים" בו תמצא רשומת היצוג למלת המפתח (אם זו הוכנסה קודם בתהליך הכנסה דומה).
משנתקבל מספר הבלוק הוא נמסר כפרמטר למכניזם המטפל בשיטת הארגון והבלוק מוכנס לזכרון.

4.1.3 ברור שאת עקרון ה-M.D (סעיף 4.) מקיימת פונקציה זו במלואה. ברוטינה קצרה בת 6 פקודות מכונה מתבצע כל חישוב מספר הבלוק הדרוש. היות ושיטת ארגון קובץ הרשומות המייצגות הוא RANDOM נעשה שימוש בתכונת המערכת לקריאה מדיסקים בבלוקים בעלי אורך קבוע. כל שיטה אלטרנטיבית אחרת הידועה בספרות כגון הטלה עבור מילת מפתח בודדת ומציאת הכתובת היחסית בדיסק, היתה מעלה את מספר הגישות לדיסק ומקטינה את הצורך לסריקה סדרתית בזכרון (כפי שנחוץ בשיטה הנבחרת על בלוק נתונים). כל עוד המערכת עומדת בתנאי צפיפות סבירים, כפי שיוסבר בהמשך, יהיה צורך רק בגישה אחת ובקריאת בלוק-נתונים אחד בלבד לזכרון.

4.1.4 דרישת ה-DATA - INDEPENDENT

עפ"י דרישה זו אנו שואפים למצב שבו הטלה של מילת מפתח כל שהיא לערך המספרי i כך שיתקיים $0 \leq i \leq 511$, תהיה רנדומית ככל האפשר ותפזר את המילים באופן זהה עפנ"י האינטרבל האפשרי. באופן כזה לא תהיינה הצטברויות בערכים מסוימים והדבר יביא למינימום חזרות. במקרה הנדון הדרישה מהפונקציה היא כזאת שבכל אחד מ-512 הבלוקים המוקצים לקובץ יהיה מספר זהה פחות או יותר של רשומות מידע דהיינו נרצה לראות האם שימוש בפונקציה מביא ליצירת ערכים מספריים המפולגים מקרית על פני הבלוקים. כדי לבדוק השערה זו סטטיסטית יש צורך להגדיר מבחן ולבצע דגימה.

4.1.5 המבחן

נלקח מדגם מקרי של 1024 מילות מפתח ועליהן הופעלה פונקצית ה-HASH. על מספרי הבלוקים שנתקבלו נערכו שני מבחנים סטטיסטיים (SMIRNOV - CHI - SQUARE, COLMOGOROV) שהראו באופן ברור שסדרת מספרי הבלוקים שהתקבלו היא רנדומית. להלן טבלת השוואה בין מספרי הבלוקים הנצפים לאלו שהתקבלו למעשה:

התפלגות מפתח "בבלוקי-נתונים"

מס' מילות-מפתח בבלוק	מס' בלוקים נצפה	מס' בלוקים שהתקבל	(נצפה) - (התקבל)
0	69	70	-1
1	138	144	-5
2	138	137	+1
3	92	88	+4
4	46	45	+1
5	18	19	-1
6	6	8	-2
7	1	0	+1
8	0	0	0
9	0	0	0
10	0	0	0

גם כאן אנו רואים שאין הבדלים משמעותיים בין מספר הבלוקים הנצפה והמתקבל בכל אחת מרמות מילות המפתח.

תנאי צפיפות המערכת

בדברנו על פונקצית ה-HASH הנבחרת הזכרנו את בעית פתרון החזרות הזהות בתוך "בלוק הנתונים" עצמו כאשר חלה גלישה מהבלוק, יותר מ-51 רשומות מידע התקבלו לבלוק. במקרה זה של גלישה (OVERFLOW) חל תהליך ההקצאה האלטרנטיבית של הבלוקים המשמשים כשטחי גלישה.
ברור שבקשת צורך לגלישה לבלוק אלטרנטיבי, כאשר יש לחפש שם מילת מפתח מוסיפה גישה נוספת לאמצעי האחסנה - הדיסק. ומגדילה עקב כך את זמן התגובה עבור אחזור מילת מפתח זו.

5 הגדרת תנאי צפיפות

ניתן לדבר על תנאי צפיפות למערכת במושגים של גלישה כפי שתוארו לעיל. ניתן להגדירם בשתי רמות שונות כתשובות לשאלות:

א. מהו מספר מילות המפתח, היחסי לנפח הכולל המוקצה, בו קיימת הסתברות משמעותית שיקרה מעבר ראשון להקצאה אלטרנטיבית של בלוק?

ב. מהו מספר מילות המפתח, היחסי לנפח המוקצה, בו מספר הבלוקים הגולשים יהיה קטן מגודל קבוע מסוים. (לדוגמא, מתי קיימת הסתברות משמעותית לגלישה של עד 51 בלוקים).

את ההרעה בתנאי צפיפות המערכת ניתן להגדיר דרך ההרעה היחסית בזמן תגובה הכללי של המערכת כאשר מתקבלת בקשה למילת מפתח הנמצאת בבלוק גלישה.
תשובה לשאלה הראשונה תקבע מהי נקודת המעבר של המערכת למצב אפשרי של גלישות, שהיא כשלעצמה עדיין איננה מהווה נקודת ציון משמעותית בהרעת תנאי הצפיפות. התשובה לשאלה השניה תקבע מהי כמות מילות המפתח הקיימת בה כדאי להעביר המערכת לנפח כולל מוקצה גדול מהנוכחי. לדוגמא מ-512 "בלוקי נתונים" ל-1024 ובכך להביא להטבה כוללת בתנאי הצפיפות.
מתוך האמור נסתכל על:

א. ההסתברות שלא יהיה אף בלוק שגלש.

ב. ההסתברות שתהיה גלישה של K בלוקים בדיוק. (K=10)
כאשר מספר מילות המפתח מגיע לאחוז מסוים ממילות מפתח הכולל להן הוקצה מקום, וההסתברות עפ"י סעיף א' לעיל קרובה ל-1 יש להתחיל לשים לב לתנאי צפיפות המערכת. כאשר עולה האחוז הנ"ל ומתקיים שההסתברות עפ"י סעיף ב' קטנה מ-0.90 נעביר את המערכת להקצאה ראשונית גדולה יותר.

5 ההסתברויות המתאימות ל-א' ול-ב' חושבו ע"ס התפלגות בינומית (וזאת עקב העובדה שהמבחנים הסטטיסטיים הראו שההבדלים בין התפלגות זו לתוצאות המדגם לא היו משמעותיים, ראה 4.1.5).
נבנו טבלאות השוואה לקיבולות שונות של מילות מפתח ונבדקו מקרי הגבול בהרעת תנאי הצפיפות ע"פ 5.1.
טבלאות ההשוואה נערכו לנפח מוקצה של:

5.2 256 "בלוקי נתונים" בהם מקום לעד 13,000 מילות מפתח. במקרה זה מתברר שבכמות מילים של 6500 (50%) חלה נקודת המעבר הראשונה. החל מ-11,500 מילים חלה נקודת המעבר בשניה (ראה דוגמאות הטבלא להלן).

5.2 512 "בלוקי נתונים" בהם מקום לעד 26000 מלות מפתח.
נקודת המעבר הראשונה החל מ-15000 מלות מפתח (57%)
נקודת המעבר השניה החל מ-21000 מלות מפתח (80%)

5.2 1024 "בלוקי-נתונים" בהם מקום לעד 52000 מלות-מפתח
נקודת המעבר הראשונה החל מ28500 מילים (55%)
נקודת המעבר השניה החל מ-40500 מילים (77%)

5.2 2048 "בלוקי-נתונים" ובהם מקום לעד 105000 מלות מפתח
נקודת המעבר הראשונה החל מ 56000 מילים (53%)
נקודת המעבר השניה החל מ 77000 מילים (74%)

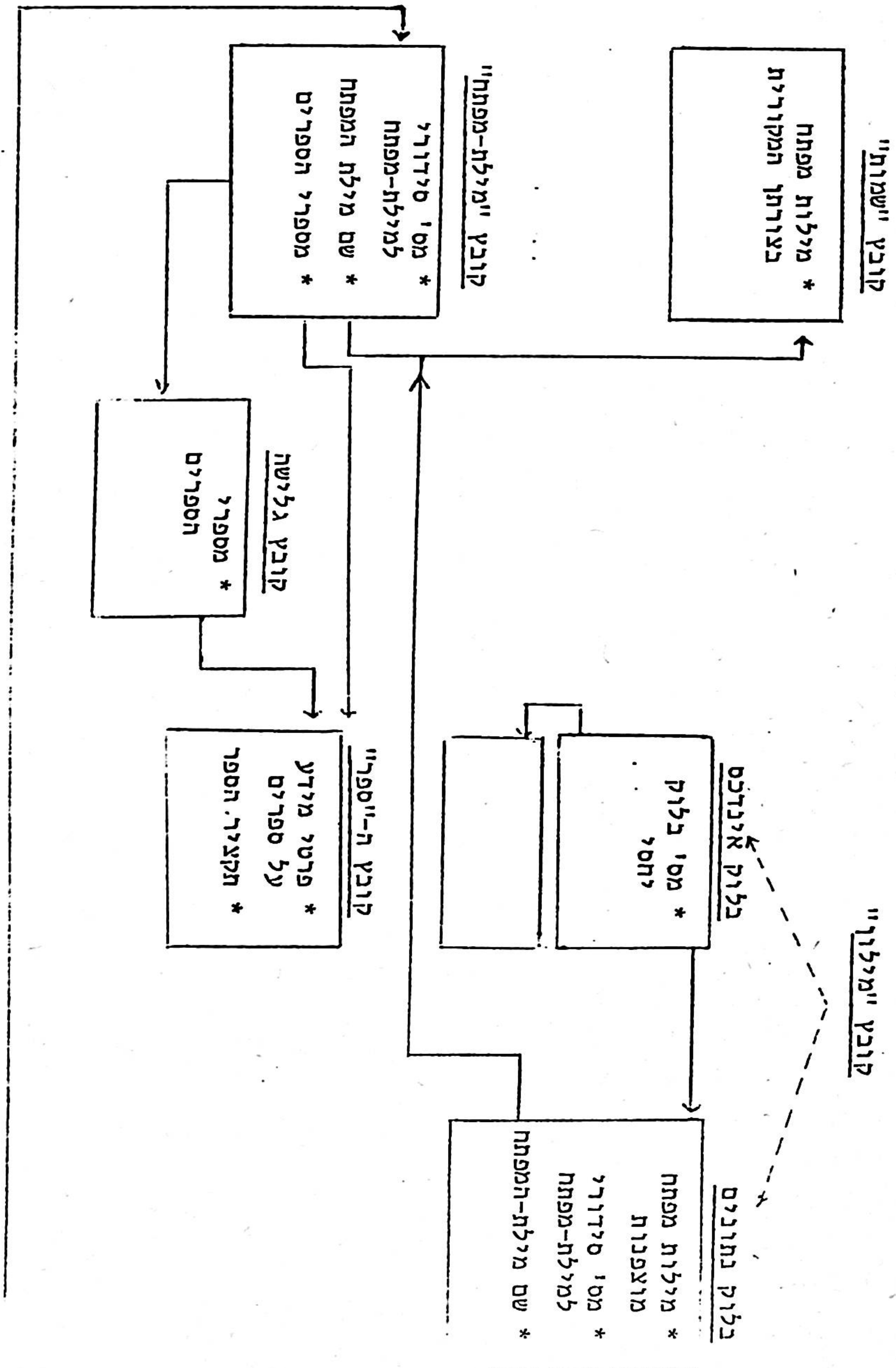
5.3 ע"ס טבלאות אלו נתנת האפשרות לתכנן את ההקצאה הרצויה ("בלוקי-נתונים") בהתאם לכמות מלות המפתח שעל המערכת להחזיק. יש לציין כמוזכר שנקודות מעבר אלו אינן קריטיות וניתן בהחלט להסתפק בזמן התגובה המתקבל גם שחלו גלישות. עקר מטרתן של הטבלאות שמכשיר עזר בהחלטה, ובציון ליתר שימת לב. להלן טבלאות השואה המתאימה ל 256 בלוקי-נתונים.

NUMBER OF ALLOCATED BLOCKS IS 256
CAPACITY OF KEYWORDS IS 13056
NUMBER OF MEGA BYTES ALLOCATED IS .125

NUMBER OF KEYWORDS	% CAPACITY OF KEYWORDS	0	1	51	+51
6500	49,7855	.997241	2.75425E-03	0	0
7000	53,6152	.993915	6.06652E-03	0	0
7500	57,4449	.974184	.025481	0	0
8000	61,2745	.898154	9.64944E-02	0	0
8500	65,1042	.680779	.26197	0	0
9000	68,9338	.310772	.364028	0	0
9500	72,7635	4.69215E-02	.144407	0	0
10000	76,5931	8.9661E-04	6.37844E-03	0	0
10500	80,4228	0	0	0	0
11000	84,2525	0	0	0	0
11500	88,0821	0	0	1.83592E-02	4.82E-02
12000	91,9118	0	0	1.37288E-02	.950532
12500	95,7414	0	0	0	1

6. מבנה סכמטי של מעברי הקבצים

להלן סכימה כללית של הקבצים הקיימים במערכת והקשרים העיקריים ביניהם.



6.1 סכום פעילות הגישה לאינפורמציה

א. חישוב פונקצית ה-HASH של מילת המפתח המבוקשת.

ב. הכנסת הבלוק המתאים (מקובץ "מילון") לזכרון ע"ס מספר הבלוק שהתקבל כתוצאה מ-(א).

ג. הפיכת מילת המפתח המבוקשת לפונקציה המוצגת (הצפנה).

ד. סריקה באמצעות אלגוריתם הדחיה של המילה המוצפנת מול בלוק הנתונים המתאים ע"פ סעיף (ב).

ה. משנמצאה מילת המפתח (פונקציות ההצפנה זהות) גישה ישירה באמצעות מספרה הסידורי לקובץ "מילות-מפתח" המהווה INVERTED-FILE ומכיל לכל מילת מפתח את מספרי הספרים המיוחסים לה. לקובץ זה המשך אפשרי בקובץ הגלישה.

ו. הוצאת מספרי הספרים עפ"י ה. והצגתם למשתמש או הפעלת פעולה בוליאנית עליהם.

ז. להקצירי הספרים המכילים פרטים אינפורמטיביים ניתן לגשת ישירות באמצעות מספרי הספרים שנשלפו ע"ס סעיף ו. להלן.

6.2 מבנה כללי של המערכת

המערכת בנויה משני חלקים עקריים המטפלים בנפרד באחזור ובקליטה. ניתן להפעיל כל חלק כזה לחוד או במקביל. פנימית בנויה המערכת מאוסף גדול של מודולים ושגרות שנכתבו כמעט בלי יחסי חפיפה ביניהם "ונתפרו" בצורה פשוטה ביותר בכל הנוגע להעברת הפרמטרים. בגלל מגבלות הזכרון נבנה חלקה העיקרי של המערכת באופן כזה שמסלול המעבר הביצועי הוא ישיר מההתחלה לסוף ללא חזרות לקטעים קודמיך וזאת כדי לאפשר מעבר לעבודה בשיטת OVERLAY ללא בעיות מיוחדות. מסיבה זו כל מודול במערכת הוא בבחינת קופסא שחורה המקבל פרמטרים ומוסרם הלאה ללא "הכרה" במודולים אחרים. כמעט אין במערכת מודול אחד המתיחס לרעהו. פילוסופית קרוב הרעיון ל-STRUCTURE-PROGRAMING.

6.3 סיכום

באמצעות מערכת זו ניתן לנהל ספריה מכל סוג שהוא ובלבד ששיטת ההתיחסות למידע מאורגנת ע"פ מילות מפתח (אינדכסים) לכך קיימות דוגמאות אפשריות רבות - ספרית תקליטים, סרטי טלביזיה, באומים, פרטיכלים של ישיבות, ספריות אוניברסיטאיות למיניהן וכדי.

THE IMPACT OF USERS ON MIS

Phillip Ein-Dor and Eli Segev
Tel-Aviv University

The users of Management Information Systems (MIS) are those members of the organization to whom the output of the system is addressed for decision making purposes. In the literature on MIS, there is a tendency to ignore the impact of users of such systems on their success or failure.

It is the purpose of this paper to present the existing knowledge on the roles of users in management information system, list the relevant variables and state propositions concerning either the relationship between the variables, or, in most cases, the relationship between the variables and the success or failure of MIS. This comprehensive view may serve as an aid to managers and systems designers, but since the number of questions overwhelms the available answers, we prefer to regard it as an index of areas requiring research and a source of hypotheses for that research.

The work described here is part of a project in which the authors have developed a paradigm of MIS incorporating the existing empirical knowledge and conceptualizations (3). In all, some one hundred variables have been identified and classified into eight major groups, namely: the MIS environment, the target MIS, the MIS project, MIS structure, executive responsibility, systems implementors, MIS users (the topic of the paper), and inter-group relations.

Following this introduction we define MIS, and MIS success, and present the variables related to the roles of users in MIS. These variables are listed in a table which summarize the findings of our study. Each of the variables is then discussed and the relevant propositions are stated. Our conclusions bring the paper to a close.

MIS and Users

A management information system is an information system that serves management rather than operations or control functions. In this view, management information systems are one of many co-equal types of information systems. Thus Esso, for example, identified six types of business information systems, one of which was management information systems. Ford also has numerous management information systems. Given this definition, a simple, stand alone, unintegrated information system serving management is a management information system.

This is a use oriented approach which states that a system is an MIS when it is used by a manager who finds it useful in the performance of his duties. The fact that information systems fall into disuse when managers change indicates that what is an MIS for one manager is not for another. Thus an MIS is an MIS for a given manager with a given problem at a given time.

We define a successful MIS as one which is profitably applied to an area of major concern to the organization, is widely used by one or more satisfied managers and improves the quality of their performance. Of these criteria, we believe that use is the most significant since a manager will use a system intensively only if it meets the other criteria. Since direct use of MIS for decision making is at present virtually confined to middle management, (see the discussion of users' rank) we regard "line managers" and "users" as synonymous, and will make no distinction between them.

The variables describing the user community are their rank, requirements, capabilities, use behavior, education, and inter-relationships with the system implementors. These variables and their attribute values which contribute to the success or failure of MIS are presented in Table 1.

Table 1
MIS user variables: success and failure sets

Independent Variable		Success set	Failure set
Variable	Factor		
requirements	identification	easy accurate	difficult inaccurate
characteristics	appropriate	present	absent
behavior	motivation to use attitudes cognitive style involvement	high favorable catered to uninhibited intensive	low unfavorable not catered to inhibited dilatory
education		appropriate	inappropriate
relationship with MIS implementors	cooperation	close	loose

Each of those variables is discussed in detail in the following section.

MIS USERS VARIABLES

Ranks of Users: In three different surveys not one instance was reported of direct use of the computer by top management. Furthermore, top management rarely receives output of any kind directly from the computer. Computers seem to have little or no effect on top management directly, but rather on the way in which middle managers contribute to top management decision making. It is possible that the level of management using the system directly is a good measur of its success.

Proposition 1: The more successful the system, the higher the level of managemen interfacing with it directly.

Users' information needs: There are two approaches to defining user requirements (1) asking the users what they need and (2) employing "expertise" and telling them what they need. The general feeling is that management needs for informati are largely ad hoc and cannot be predicted by even the most skillful managers.

Nevertheless, it is often assumed that user needs can be defined in advance These seemingly conflicting notions can be resolved by the explanation that the higher one goes in the organizational hierarchy decisions become less structured and are amenable neither to modelling not to algorithmic solutions, let alone definition of data requirements.

Difficulties also arise from a lack of understanding of management's wants, from the physical inaccessibility of managers to those wishing to ascertain thei needs and from hesitancy on the part of some managers to formally identify the criteria they wish used in decision making. Finally, doubts have been raised that even if managers define their information requirements, that they really need the information they want.

Proposition 2: The lower the level of management, the more easily information needs can be identified.

Proposition 3: The more easily management needs for information can be accuratel identified in advance, the greater the likelihood of success.

User characteristics and capabilities: In successful system efforts, operating managers participate who are well motivated, with general expertise in the functional area, aware of the major sources of information and alternative method

of supplying data; equipped with some knowledge of computer capabilities; esponsive and responsible. Conversely, development has been retarded by lack of appreciation or education of many managers concerning computer systems, lack of skills, techniques and controls and by a perception of the MIS as a threat to control over the work environment.

Proposition 4: A lack of appropraite traits in system users contributes to system failures.

Use behavior: Motivation. This variable has to do with converting potential into active users. The assumption that the system itself will provide the motivation (since the user will be working with better data in more useful forms and this will increase his confidence), is contradicted by empirical findings. Lack of use of computers for important decisions by top managers is sometimes attributed to lack of motivation: "There is much desire but little passion".

The key to motivation may lie in the rewards system. It was found that paid subjects in an experiment realized significantly greater profits and used more decision time thanunpaid subjects.

Proposition 5: Managers will be motivated to use the system if there are rewards associated with so doing.

Proposition 6: The greater the motivation to use the system, the higher the level of use.

Attitudes. Swanson (9) found experimentally that a priori involvement of a user with MIS will increase his MIS appreciation and that use involvement will also increase appreciation. Likewise, Lucas (7) reports that favorable user attitudes are associated with use. Guthrie's survey (4), however, found that the direction of change in attitude depends on the user's experience. Thus he reports that it is clear that good experience increases the managers' awareness of the need for better information and very positively influences their concepts of the impacts of MIS development; while negative attitudes are due to many unfortunate experiences with hardware oriented "so called" MIS imposed on users by data processing specialists. (This is supported by Lucas' (7) report that poor system quality leads to unfavorable user attitudes)

There seem to be heavy emotional and psychological costs associated with the changes caused by information systems development: 1) a central theme running through discussions with managers was a concern for the impact of MIS on the people in the organization; 2) the introduction of a computer system causes significant attitudinal changes, which can adversely affect performance; 3) in a system poorly introduced, individuals did not know their new functions in the system, and its benefits were not communicated; as a result, while there was no malicious resistance, confusion and apprehension reigned with employee reaction ranging from fear of job down-grading and break up of work relationships, through perceived threats to self-esteem, to totally negative views of the imposed changes. 4) In medical systems it has been reported that when automation requires that habit patterns be changed, physicians and nurses, have been reluctant to do so.

Argyris (1) studies the effects on management of MIS development and found it to be a stressful and emotional problem to the participants; reduction of freedom of action, psychological failure, changes in success criteria and decreased feelings of essentiality were identified as tending to create genuine resistance. On the other hand, Guthrie (4) did not find support for the notion of managerial aprehension over MIS and concluded that "massive resistance" by user managers does not appear to be a general constraint to MIS development.

Proposition 7: Favorable user attitudes lead to increase use of MIS.

Proposition 8: User attitudes become more favorable with positive personal and organizational experience and more unfavorable with negative experience.

Cognitive style. It has been noted that top management decision making is highly intuitive, very personal and informal. Thus it has been posited that individuals with different cognitive styles and different uses and different levels of information systems. As a consequence, many systems produce information that is perceived as useless by those receiving it. A successful case report, on the other hand, attributes success to the use of a computer to serve individual requests and so provide managers with information meaningful to them. Hammond (5) suggest that considerable scope should be allowed for judgemental inputs to models. In general, systems must accomodate management needs, otherwise management will not use them. Cases are quoted of administrators unwilling to use computer systems due to intuition-based decision habits.

Proposition 9: An information system will be used most by those managers to whose cognitive style it is best adapted.

Proposition 10: The less inhibited users are towards information systems the higher the level of use.

Involvement. The involvement of operating personnel has been found in field studies to contribute to successful implementation and its absence to failure. Part of the explanation may lie in the observation that appropriately qualified operating managers are likely to be a better source of ideas for profitable changes in operations than are computer professionals. Areas in which involvement has been considered desirable include:

project selection
project definition
establishing design criteria
project planning and design
responsibility for functional specification
project manning
responsibility for implemantation progress and results
establishing operating criteria

Acceptance of the system by the users is conditioned on the user's feeling that he, not his delegates, contributed to its development. However, a major problem with attaining user involvement is the amount of time required. The involvement of ultimate users must be managed judiciously so as not to consume too much of their time. As an example of the magnitude involved, Carlson (2) notes that, in the case he reports, the chairman of the project steering committee devoted up to a quarter of his time to following the working group's activities during a critical phase.

User/managers cannot be expected to give information systems development significant time and effort. Perhaps managers want to have a lot of influence in the MIS design and implementation, but are unwilling to devote the study, time and effort required to make their participation meaningful. In practice there is a widespread tendency for users to abdicate their responsibilities in system design to the computer specialists.

Proposition 11: User involvement in system development will be facilitated if the amount of time required is kept within bounds and if the involvement is close to the user's jurisdiction.

Proposition 12: The higher the level of user participation in system design and implementation the higher the level of use.

User-implementator relations: The relationship between managers/users and MIS implementors seems to be fraught with difficulties, which may lead to conflict between these groups. Such conflict may prevent the success of MIS projects. Conflict provoking factors which have been identified in this relationship include faculty communications relocation of power and implementation practices.

Communications. There are numerous reports of an unbridged culture and communications gap between implementors and users. Computer technicians are complicators, not simplifiers and tend to use jargon which confuses, intimidates or angers users. Thus management is often confused by technological arguments between systems people. This gap has been attributed to different styles of thinking and problem solving in the two groups.

Relocation of Power. As a gradual transfer of power has taken place to the information services department, users have become more powerless and more frustrated, the frustration being reflected in unfavorable attitudes toward the information system and its staff.

MIS implies an organizational design in which competence and cooperation become more important than power and competition and the change tends to create fears and resistance on the part of individuals, groups and intergroups. It has been reported that the middle manager at the staff level has recurring nightmares about being displaced by a computer. This situation may create an atmosphere conducive to rumors or false information which are overblown, initiate negative sentiments and foster conflict.

Conflict. User interaction with implementors is most noticeable at the systems application and programming stage which involves the transfer of system plans and policies into an operating environment. The unfavorable user attitudes generated by conflict can lead to mutual unwillingness to cooperate, a refusal on the part of users to participate in design and operation of the information system, and rejection by technicians of good ideas which may be generated by line managers. Resistance to change can complicate the design process and takes on the form of quitting, open opposition, apathy and quiet destructiveness. The results are unresponsive and uncreative work behavior, uncertainty, anxiety and even a surprising amount of sabotage.

The manner in which the information system staff go about their tasks can also affect their relationship with users. Frequent changes of systems without consulting users can adversely affect their attitudes. Gibson & Nolan (22), predict that the information chief is bound to be criticized by users either for allowing stagnation in the name of stability or for introducing change in the name of progress and the state of the art. In general, the systems design and operating policies of the information services department and technical quality of the systems influence user attitudes and perceptions of information systems staff.

Lack of cooperation between those two groups is especially critical since neither the manager nor the analyst alone can design and implement a true MIS; cooperative teamwork and dialogue are essential. Murdick & Ross (8) recommend the involvement appropriately qualified managers as a means of bridging the communications gap. Means to attaining such involvement include increased sensitivity of implementors to users, education, role definition and project formalization. If user involvement is achieved, it results in favorable user attitudes and perceptions of the information system and its staff.

Sensitivity to users. Many writers place the onus for attaining cooperation

on the implementors. To help the user, the implementor must develop an understanding of the user's work and information. Thus it is the implementor who should take the first step in bridging the communications gap.

Successful systems men learn that they must take managers' attitudes and preference into account. They must also accomodate and support users' more intuitive style. Among other things, this will be reflected in basing some decisions about the importance of information on the relative importance of the manager responsible for the information. Argyris notes that MIS people, who presumably possess the requisite technical qualifications, have the same emotional problems with each other as they have with line managers. The strategy suggested by Argyris is to raise the interpersonal compentence of both managers and MIS personnel in dealing with emotionality and strain in interpersonal and intergroup problems. If interpersonal skills are improved, he believes that they will naturally turn to education and structural change.

Role definition. It has been recommended that status incongruence between users and implementors be avoided, that users and implementors should share status and background, and that their roles be clearly defined: planners making recommendations, line managers making decisions and implementors making information systems.

Formalization. Performance standards and short term control devices could be formalized for the more routine tasks and remove the MIS personnel involved from frequent interaction with the users. Without project management guidelines, implementors and users will have to develop their own procedures for dealing with each other.

Proposition 14: The better the communications between the MIS staff and operating management, the higher the level of user involvement.

Proposition 15: The lower the level of conflictbetween MIS staff and users, the higher the level of user involvement.

Proposition 16: The more developed the interpersonal skills of users and implementors, the lower the level of conflict.

Proposition 17: The higher the level of user involvement, the greater the degree of cooperation between users and implementors, the greater the likehood of success.

Summary and Conclusions

Since we regard an MIS as successful when it is widely used, the user becomes, in fact, an important element of the system. This approach may lead to different conclusions concerning system design and development strategy than would the now prevalent approach based in large part on technological rather than behavioral criteria.

Probably the most critical variables are users' behavior and their relationship with implementors. Little, however, is known about these variables and considerable research is called for. First there is a need to develop measures which would make it possible to ascertain the values of the parameters for a given organization. Secondly, criteria are required to distinguish the success set values from those contributing to failure. Finally, methods are needed which would enable an organization which finds itself predominantly in the failure set to move into the success set and to prevent organizations in the success set from retrogressing into failure.

REFERENCES[1]

(1) Argyris, Chris. "Management Information Systems: The Challenge to Rationality and Emotionality". Management Science. Vol 17, No. 6, (February 1971), pp. B-275-B-291.

(2) Carlson, Walter M. "A Management Information System Designed by Managers". Datamation, Vol. 13, No. 5 (May 1967), pp. 37-43.

(3) Ein-Dor, Phillip & Segev, Eli, "A Paradigm for MIS". Tel-Aviv University, Faculty of Management, The Leon Recanati Graduate School of Business Administration. Working Paper 256/76 (May 1976).

(4) Guthrie, Art. "Attitudes of the User-Manager Towards Management Information Systems". Management Informatics, Vol. 3, No. 5 (1974) pp. 221-232.

(5) Hammond, John S. III, "Do's and Don'ts of Computer Models for Planning". Harvard Business Review, Vol. 52, No. 2 (March-April 1974), pp. 110-123.

(6) Heany, Donald F. "Education: the Critical Link in Getting Managers to Use Management Systems". Interfaces, Vol. 2, No. 3 (May 1972) pp. 1-7.

(7) Lucas, Henry C. Jr. "Systems Quality, User Reaction and the Use of Information Systems". Management Informatics, Vol. 3, No. 4 (August 1974), pp. 207-212.

(8) Murdick, Robert G. & Ross, Joel E. "Future Management Information Systems, Part II". Journal of Systems Management, Vol. 23, No. 5 (May 1972), pp. 32-35.

(9) Swanson, E. Burton, "Management Information Systems: Appreciation and Involvement". Management Science, Vol. 21, No. 2 (October 1974), pp. 178-188.

(1) Due to space limitations, the reference list has been shortened to a necessary minimum. A detailed list is available from the authors.

STRUCTURE AND THE SUCCESS OF MANAGEMENT INFORMATION SYSTEMS

Phillip Ein-Dor and Eli Segev

Tel-Aviv University

Introduction

The work reported in this paper is part of a project in which the authors have developed a paradigm of MIS incorporating the existing empirical knowledge and conceptualizations in the field (3). The project consisted of a comprehensive scan of the literature, identification of the relevant variables, their classification into major groups and the formulation of propositions concerning interrelationships between the variables and their influence on the success or failure of MIS.

In all, some one hundred variables have been identified and classified into eight major groups, namely: the MIS environment, the target MIS, the MIS project, MIS structure, executive responsibility, system implementors, MIS users and inter-group relations.

This paper synthesizes the existing knowledge on the structural variables of MIS and states over twenty propositions concerning their interrelationships and impact on success or failure.

Success and failure of MIS

Indicators of the success of MIS include:

- profitability,
- application to major problems of the organization,
- quality of decisions or high performance,
- user satisfaction,
- widespread use.

We then define a successful MIS as one which is profitably applied to an area of major concern to the organization, is widely used by one or more satisfied managers and improves the quality of their performance.

We believe that use is the most significant of these criteria, since a manager will use a system intensively only if it meets the other criteria as well.

MIS Structure

An early study suggested a schema for evaluating information systems in terms of system characteristics, individual functional fit, corporate functional fit, horizontal fit (coordination of functions) and vertical fit (upward and downward flow of information and control). This characterization partly overlaps the definition of an information system by Mason and Mitroff (11) as "a PERSON of a certain PSYCHOLOGICAL TYPE who faces a PROBLEM within some ORGANIZATIONAL CONTEXT for which he needs EVIDENCE to arrive at a solution, where the evidence is made available through some MODE OF PRESENTATION".

Although we shall not consistently adopt the internal taxonomies of either of these schema, we will adopt their general approach and present the available evidence on information system structure in terms of system characteristics, individual fit, user interface and organizational fit.

These variables and the detailed list of their various factors are exhibited in Table 1. This table is both a guide to the variables discussed here and a summary of the propositions concerning their direct impact on success and failure.

Table 1
Summary of Structural Variables of MIS
Success and Failure Sets

Independent variable		Success	Failure
Variable	Factor	Set	Set
system characteristics	data characteristics	appropriate	inappropriate
	data base:	existent	non-existent
	- capabilities	appropriate	inappropriate
	- development strategy	appropriate	inappropriate
	- data elements	meet needs	do not meet needs
	processing capabilities	highly developed	underdeveloped
		flexible	inflexible
	integration	high degree	low degree
individual fit	cognitive style	good fit	poor fit
	system adaptability	high	low
user interface	I/O characteristics	suitable	unsuitable
	mode of operation:		
	- OLRT	implemented	not implemented
	- direct access	available	not available
	- complexity	low	high
	- technical specifications	human engineered	not
	- time sharing	implemented or not	implemented or not
	- mode of communication	favored	disfavored
organizational fit	feasibility	feasible	infeasible
	integration	high degree	low degree
	information structure	mirrors organization	does not
	information system	reflects <u>weltansicht</u>	does not

System Characteristics

System characteristics are determined by: 1) data characteristics, 2) the data base, 3) processing capabilities and 4) integration of the information processing system. Each of these factors is discussed below.

Data characteristics. Data in a system can be characterized by a number of attributes, the five most prominent among them being:

- accuracy and measurement error,
- content or relevancy,
- recency or information delay,
- frequency,
- feedback time - the time elapsing between implementation of a decision and the receipt of a report on the effect of the decision on the environment.

Proposition 1: Appropriate data characteristics enhance the likelihood of MIS success.

The data base. A data base may be defined as the physical representation of one or more logical data sets and a mechanism for retrieving extracts of the representation and for changing the representation as the logical data change.

It is generally conceded that the usefulness of an MIS depends critically on a data base, four aspects of which are discussed, namely its necessity, its capabilities and development and content strategies.

1. Data base necessity. A variety of factors make the data base an indispensable component of the MIS. Among these:

- data integrity. The data base assists in keeping data up to date, minimizes redundancy, and reduces lack of data element standardization as a reason for bad reports.
- responsiveness. The data base, as a means of structuring data, maximizes accessibility of data and facilities response to ad hoc management requests with reasonable dispatch and efficiency.
- data independence. The concentration of data in a corporate data base permits the separation of data from specific applications and the sharing of data by different programs.
- decision support. The data base supports the development of decision models by providing the historical data necessary to develop relationships between variables.

Proposition 2: The existence of a data base is critical to the success of an MIS.

2. Data base capabilities. The capabilities required of a data base management system are:

- file design, creation, maintenance and update,
- report generation,
- information retrieval,
- on line inquiry.

These capabilities are embodied in appropriate hardware and software elements including the computer, storage devices, a communications network, a volume structure of information, data base organization methods, languages for information retrieval and software for accessing the data base.

Proposition 3: Appropriate data base capabilities are essential to the success of an MIS.

3. Data base development. A strategic decision on which opinions are divided concerns the question whether existing files should be built on and transformed

into the data base or should they continue undisturbed while the data base is built that will eventually replace them?

4. Content. A second strategic decision concerns the content of the data base and arises from the infeasibility of analyzing management information needs, since these are unknown. Theoretically, the solution is to store all data elements which may possibly be required; this solution is unrealistic because of excessive data base size and retrieval time. This forces designers to compromise between prestructuring data, for ease and rapidity of retrieval, versus redundancy, to facilitate unanticipated forms of search.

Proposition 4: The optimality of a strategy for data base construction is dependent on the organizational situation.

Proposition 5: The likelihood of MIS success is enhanced by selection of an appropriate data base construction strategy.

Proposition 6: The better the fit between user needs and data elements included in the computerized data base, the greater the likelihood of MIS success.

Processing capabilities. These are the data manipulation and data reduction capabilities of the system. The need for data reduction is sparked by cognitive limitations which, in situations of information overload, lead to less information being used as more and more is provided. Many computerized information systems tend to produce massive amounts of data which induce precisely this condition of information overload.

There are conflicting theses and empirical findings with respect to the relationship between the level of aggregation and user behavior. Mock (13) found that subjects in an experiment who received more detailed feedback performed better than subjects with summary feedback information. In a different experiment, Chervany and Dickson (1) report that decision makers given simple summary data 1) made higher quality decisions than those receiving the same data in standard formats, 2) had less confidence in the quality of their decisions and 3) took more time to make their decisions. A resolution to these apparently conflicting empirical findings seems to lie in the observation that there is an amount of information which is optimal for the decision maker.

Data reduction consists not only of aggregation, but also of filtering. The filter should be designed so that, on the one hand, each level and position of management is provided with all the data than can be used in the conduct of its tasks; on the other hand, only information that can and must be acted on is received at each managerial level and position.

Proposition 7: The more highly developed and the more flexible the processing capabilities of an MIS, the greater the likelihood of success.

Integration. Integration is a frequently mentioned characteristic of MIS. It has, however, at least five different aspects:

- semantic integration at the individual level. This blends separate pieces of data into a single coordinated image,
- integration of data from different areas of the corporation by means of a data base,
- integration of models. Integration of tactical and strategic models, the output of one model being mapped into or becoming input to another,
- integration of models with the data base. This integration is implemented as the models draw the majority of their inputs from the data base and place their outputs back into it,

output integration. King and Cleland (7) see integration in use, rather than process; the outputs of the various subsystems should be compatible and available in the aggregate.

roposition 8: The greater the degree of data integration, the greater the likelihood of MIS success.

ndividual Fit

Individual fit, the second of the four structural variables of MIS, refers to ne extent to which different people need different information and the means for dapting information to people.

ypology of users. Several writers have noted that decision makers may be divided nto groups distinguished by their information gathering and evaluation haracteristics.

It has been shown, in experimental situations, that the decision or information valuation style is a significant determinant of behavior. Thus a central factor etermining whether a manager will use a model depends on the extent to which it its his cognitive style.

Since different people look at problems in different ways and require different inds of information and output formats, Hayes & Nolan (4) raise the question whether he concept of a general model which can be used by a number of people for different urposes over a long period of time is at all valid.

ailorizing. The behavioral dependence of information systems suggests the validity f tailorizing them for users. A tailorizing capability would be reflected in the daptability and coverage of the system.

It is widely agreed that to support the changing needs of different users, ystems should be responsive and flexible. Adaptation to managers' needs implies the bility to change the system readily and this implies an interactive capability, iscussed in the next section.

Coverage refers to the subset of the information base which is made available o a user. On the one hand, a simple data processing system fails to give management complete picture. On the other hand, large, all inclusive models overwhelm the anager's ability to understand the underlying assumptions and to integrate the utputs into the decision process.

There are however, costs associated with fitting information structures to users. olland et al., (6) report that users have very strong possessive feelings about their" data and are reluctant to share them. Often the original user of a model is he only one with sufficient understanding and confidence in it to use it, and the odel falls into disuse with his departure.

roposition 9: The better the fit of a system to a manager's cognitive style, the greater the likelihood that he will use it.

roposition 10: The more adaptable a system to different managers and different needs, the greater the likelihood of success.

he User Interface

The user interface, the third structural variable of MIS, consists of two actors - input/output characteristics (the user-report interface), and mode of peration (the user-machine interface).

nput/output characteristics. These refer to the symbols and formats by means of hich inputs and outputs are represented physically. These should be simple, amiliar and easily understood. An unsuitable symbol set and poor formats are the ajor ingredients of unsatisfactory reports. Kneitel (8) advocates the presentation f data in such a way as to attract the attention and interest of the user. An xperiment supporting this approach found that subjects showed marked preferences

and reciptivity for information that is embedded within a story or character laden context rather than purely impersonal information (12).

Proposition 11: Suitable output symbols, formats and contexts enhance the likelihood of MIS success.

Mode of operation. A sharp dividing line is recognized between the two basic modes of operation - batch processing, which produces data on a deferred basis, and on-line-real-time (ORLT) which provides immediate response. The dichotomy expresses itself in cost and system complexity, in the technical specifications of the system and in the techniques used for its implementation. Four aspects of the mode of operation are discussed below: OLRT application, technical specification, problem solver's behavior and mode of communication.

1. OLRT application. Sackman (16) found that, in problem solving situations, timesharing leads to a higher quality product and faster problem solving; in the case of Xerox (Canada), timeshared models have proven successful in assisting in corporate planning (17).

It has been stated and validated in at least one case (17), that an investigation of the manager's machine interface may possibly provide a solution to the information overload problem. Furthermore, interactive systems may provide managers with direct access at a speed and convenience congruent with their decision making process.

Proposition 12: The implementation of on-line-real-time systems enhances the likelihood of MIS success.

Proposition 13: Direct access to the system enhances management use.

Proposition 14: The more complex the mode of operation chosen, the greater the likelihood of failure.

Proposition 15: The more successful the MIS, the more extensive its direct use by management.

2. Technical specifications. Technical specifications of the system - determining the operating characteristics realized at the terminal device employed for input/output - and the human engineering factors built into the device may be critical to success. User reactions have been found to depend on 1) response time, 2) verbosity or terseness, 3) isolation and 4) computer expertise. Head (5) states, from a human factors standpoint, a system response time of three to five seconds on the average is desirable.

Proposition 16: Technical specifications with a human engineering orientation enhance the likelihood of MIS success.

3. Behavior. The behavior of problem solvers in time sharing environments and in comparison with batch processing environments has been studied relatively extensively. Summarizing five studies comparing time sharing with batch processing, Sackman (16) concluded that there is a 20 percent man hour advantage in favor of time sharing while 40 percent more computer time is used in time sharing than in batch. When only computer time and man hours are taken into account, the tradeoff is balanced and neither system is more economical than the other.

The reaction of computer dependent workers to on-line-systems has also been recorded in a study of the effect of terminal use on the motivation of bank tellers which found that the use of terminals had no effect on absenteeism or feelings about the job. On the other hand, terminals were associated with more positive feelings about co-workers and responsibility for quality work and these feelings became more positive with longer experience. There was no change in feelings about the work itself and chances for development and a moderate decrease in sense of accomplishment.

In Sackman's opinion, the main behavioral result of on-line-off line studies the existence of very large differences in human performance, differences which ar typically an order of magnitude larger than computer system differences. The cost effective payoff thus lies more in human factors than in machine or system configuration variables. This conclusion is supported by the experimental findings reported by Cohen & Van Horn (2).

Finally, Sackman reports that users spend most of their time away from the terminal. Thus man-computer problem solving consists mainly of introspection or man-to-man communication, and major insights were observed to occur away from the terminal. It is a human directed process with the user as the central source of strategic intelligence and the computer as a tactical aid. It is misleading, therefore, to refer to man-machine symbiosis in time shared problem solving.

Proposition 17: The use of time sharing systems improves problem solving performa

Proposition 18: The implementation of time sharing is not critical to MIS success

4. Mode of communication. The mode of communication is a final aspect of the us interface which proves to be significant. Ochsman and Chapanis (14) report a sharp dichototmy between modes of communication involving voice, and those that do not. There is also considerable evidence that managers specifically favor verbal channe as a means of supplementing and even replacing formal sources of information. Few systems to date have implemented audio response devices.

Proposition 19: The implementation of communication modes which are favored by managers enhances the likelihood of MIS success.

Organizational Fit

The fourth and the last structural variable of MIS is organizational fit. The are two aspects to the organizational fit of an information system. One aspect has do with the readiness of the organization to absorb an information system; the oth aspect is the extent to which an information system is adapted to the particular organization in which it is embedded, the subject of this section. Very little wor seems to have been done on this topic directly, but there are some more or less oblique references in terms of systems feasibility, integration, the organizationa interface, and weltansicht.

Feasibility. The feasibility of an information system is regarded as consisting o technical, economic and operational or organizational aspects.

The technical aspect deals with the ability of the system to handle problems relevant to the organization. The economic aspect has to do with the economic valu of information and the cost of producing the information. The organizational aspec deals with the degree to which the organization will be ready to accept and use a particular information system.

Proposition 20: Technical, economic and organizational feasibility are prerequisi for MIS success.

Integration. Integration, as it refers to corporate fit, has to do with the integration of organizational activities. Thus Schwartz (16) recommends that busin transaction systems, management information systems and planning and control syste be planned on a coordinated and unified corporate wide basis. McFarlan (10) refers the integration of MIS objectives and plans in general corporate objectives. This an issue that arises in the design stage.

Proposition 21: The greater the degree of organizational integration the greater the likelihood of MIS success.

Organizational interface. Information systems exist within an organizational con and a major problem is the introduction of available technology into that environm

This requires solutions to problems of organizational behavior in planning, design, implementation and organization. It has been noted that the information structure is a function of the organization's needs and consequently should reflect the overall corporate character and design. Lucas (9) finds that information services have a high score on all variables in the organizational power model and potential for high scores on all variables in the conflict model. This may explain the high propensity for organizations to reject information system implants.

Proposition 22: The more closely the information structure reflects organizational structure, the greater the likelihood of MIS success.

Weltansicht. Finally, an information system fits the organization to the extent that it reflects its weltansicht or view of the world. Hayes & Nolan (4) state that it is less important that models contain a complete or correct representation of reality but they should reflect the company's own understanding of the reality. Mitroff, Nelson & Mason (12) state that information is information if and only if it is tied to an appropriate story that has meaning to the user and to the organization in which he is located.

Proposition 23: The more closely the information system reflects the organization's view of the world, the greater the likelihood of success.

Conclusions

The table at the beginning of the paper summarizes our propositions concerning the direct effect of structural variables on MIS success. It is evident from the table that while appropriate technical characteristics of the system are a necessary condition, they are by no means sufficient. In order for an MIS to succeed it must be adapted to the idiosyncracies of the organization it serves and of the individuals comprising it. It appears however, that the technical factors have been studied much more extensively than the organizational and behavioral factors. Much research needs to be devoted to the organizational interface of the technology and to the identification of methods which will ensure a good fit along that interface.

References

(Note. Due to space restrictions, the reference list has been reduced to an essential, but representative, minimum. The authors will be happy to provide a complete list on request).

1. Chervany, Norman L. & Dickson, Gary W. "An Experimental Evaluation of Information Overload in a Production Environment". Management Science, Vol. 20, No. 10 (June 1974), pp. 1335-1344.

2. Cohen, I.K. & Van Horn, Richard L. "A Laboratory Research Approach to Organizational Design". European Institute for Advanced Studies in Management. Working Paper 72-16 (April 1972).

3. Ein-Dor, Phillip & Segev, Eli. "A Paradigm for MIS". Tel-Aviv University, Faculty of Management, The Leon Recanati Graduate School of Business Administration. Working Paper 256/76 (May 1976).

4. Hayes, Robert H. & Nolan, Richard L. "What Kind of Corporate Modeling Functions Best". Harvard Business Review, Vol. 52, No. 3 (May-June 1974), pp. 102-112.

5. Head Robert V. "Real Time Applications". Journal of Systems Management, Vol. 25, No. 9 (September 1974), pp. 7-13.

6. Holland, W.E., Kretlow, William J. & Ligon, Jerry C. "Socio-Technical Aspects of MIS". Journal of Systems Management, Vol. 25, No. 2 (February 1974), pp. 14-16.

7. King, W.R. & Cleland, D.I. "Environment Information Systems for Strategic Marketing Planning". Journal of Marketing, Vol. 38, No. 4 (October 1974), pp. 35-40.

8. Kneitel, Arnold M. "Will the Real Users Please Stand Up". Management Datamatics Vol. 4, No. 3 (June 1975), pp. 83-93.

9. Lucas, Henry C., Jr. "A Descriptive Model of Information Systems in the Context of the Organization". Data Base, Vol. 5, Nos. 2, 3, & 4 (Winter, 1973), pp. 27-39.

10. McFarlan, F. Warren. "Problems in Planning the Information System". Harvard Business Review, Vol. 49, No. 2 (March-April 1971), pp. 75-89.

11. Mason, R.O. & Mitroff, I.I. "A Program for Research on Management Information Systems". Management Science, Vol. 19, No. 5 (January 1973), pp. 475-487.

12. Mitroff, Ian I., Nelson, John & Mason, Richard O. "On Management Myth-Information Systems". Management Science, Vol. 21, No. 4 (December 1974), pp. 371-382.

13. Mock, Theodore F. "A Longitudinal Study of Some Information Structure Alternatives". Data Base, Vol. 5, Nos. 2, 3 & 4 (Winter 1973), pp. 40-49.

14. Ochsman, Robert B. and Chapanis, Alphonse. "The Effects of 10 Communication Modes on the Behavior of Teams during Co-operative Problem-Solving". International Journal of Man-Machine Studies, Vol. 6, (1974), pp. 579-619.

15. Sackman, H. "Experimental Analysis of Human Behavior in Time Sharing and Batch Processing Information Systems" in Kriebel, C.K. Van Horn, R.L. & Heames, T.J. (eds.) Management Information Systems: Progress and Perspectives. Pittsburgh: Carnegie Press, (1971).

16. Schwartz, M.K. "MIS Planning" Datamation Vol. 16, No. 10 (September 1970), pp. 28-31.

17. Seaberg, Ronald, A. & Seaberg, Charlotte. "Computer Based Decision Systems in Xerox Corporate Planning". Management Science, Vol. 20, No. 4, Part II (December 1973), pp. 575-584.

קבוצת דיון מס' 2:

"יישומי מחשבים בהנדסה ובתעשיה"

יו"ר: פרופ' א. שמיר

י. לאב

ד. ערמון

מ. כפיר

מגמות חדשות ביישום מחשבים בתעשייה

ד"ר יונה צ. לאב
המכון לאוטומציה תעשייתית
המכון לפריון העבודה והייצור
תל-אביב

ת ק צ י ר

הפיתוח המזורז של מחשבי מיקרו בשלוש השנים האחרונות פתח אופקים חדשים לשילוב מחשבים בבקרה של מרבית התהליכים התעשייתים ומכונות הייצור. הפיתוח איפשר ממוש רעיונות שפותחו בתחילת שנות הששים בתחום הבקרה הספרתית הישירה (DDC) ובתכנון מערכות בקרה ממוחשבות היראכיות ומפולגות. במאמר נסקרות שיטות קיימות ביישום מחשבים במערכות בקרה תעשיתיות ומנותחות המגמות החדשות בתכנון המערכות, בייצור ציוד מודולרי וסטנדרטי, ובפיתוח תוכנה סטנדרטית. בחלקו האחרון של המאמר מתוארות מערכות שיושמו בהצלחה בעולם ובישראל, ונסקרים המאמצים הדרושים כאן לשילוב נרחב של הטכנולוגיה החדישה.

מ ב ו א

נסיונות ראשונים לפיתוח מערכות-מבוקרות-מחשב (computer controlled systems) נעשו כבר באמצע שנות החמישים[1] ומאז שולבו מחשבי מיני רבים במערכות תעשיתיות, אך רק בשנתיים האחרונות נוצרו התנאים לשילוב מעשי וזול של מחשבים במכונות ובתהליכי הייצור. ההתפתחות התאפשרה עם תחילת ייצורם הזול וההמוני של מחשבים ספרתיים, במיוחד של מיקרו מחשבים ומעגלים אלקטרוניים מסולבים המיוצרים בשיטות ה- LSI ובשיטות היברידיות. ההתפתחויות המהירות הצפויות ביישומם של מחשבים בתעשיה מבוססות על מחקרים רבים, שנעשו בעשרים השנים האחרונות, ועל הנסיון הרב שנרכש בתקופה זאת בתכנון ובהפעלת מערכות ממוחשבות בתעשייה.

מטרת מאמר זה לנתח את המגמות הצפויות בשילוב המחשבים החדישים בתהליכי הייצור. בדיון לא כלולות הבעיות הכרוכות בעיבוד נתונים מינהלי בתעשייה. המאמר מחולק לארבעה חלקים: בחלקו הראשון נסקרים המושגים, השיטות ותחומי היישום שהיו מקובלים בתחום הנידון, וכן מפורטות חלק מהסיבות שהקשו על שילובם הנרחב של המחשבים עד כה. בחלקו השני מנותחות המגמות הצפויות בשיטות שילובם של מחשבים בתהליכי הייצור שהתאפשרו כתוצאה מהתפתחות המיקרו-מחשבים. בחלקו השלישי מתוארים מערכות ויישומים טיפוסיים שפותחו לאחרונה בחו"ל. לבסוף תתוארנה מערכות שפותחו ויושמו בהצלחה בישראל והמאמצים הדרושים כאן לשילוב נרחב של הטכנולוגיה החדישה.

מחשבים מנוצלים בתהליכי הייצור בביצוע ארבע סוגי פעולות עיקריות: איסוף נתונים, השגחה, פיקוד ובקרה אוטומטית. - כל אחת מהפעולות הללו נתנת לממוש בדמות שונות, ולמעשה, ניתן למכנן וליישם היראכיה של מערכות פיקוד ובקרה בכל מתקן ומפעל. ברמה הנמוכה ביותר נאספים נתונים ומבוצעות פעולות בקרה של משתנה יחיד במעגל הבקרה בתהליך הייצור, כגון: מדידת טמפרטורה, מעקב אחר לחץ חריב במיכל מסוים, או בקרת מהירות סיבוב של מנוע. פעולות המדידה והבקרה ברמה זו מבוצעות בתדירות גבוהה ומורכבות החישובים היא קטנה. אלגוריתמי הבקרה המיושמים ברמה זאת מבוססים לרוב על השיטות הקלסיות שפותחו בתורת הבקרה האוטומטית עבור מערכות עם משתנה יחיד. ברמה השניה מבוצעות הפעולות הללו בתת-מערכות שלמות במערך הייצור, כגון, בקרה בו-זמנית של התנועות במספר צירים במכונה לעיבוד שבבי. הפעולות ברמה זאת מחייבות טיפול במערכות רבות-משתנים, ולעיתים, מבוצעים ברמה זאת חישובים לקביעת נקודת העבודה האופטימלית של כל אחד מהמשתנים הבודדים. נקודת העבודה המחושבת נמסרת לרמה הנמוכה יותר, בה נאספים הנתונים ובה מבוקר המשתנה הבודד. ברמה השלישית מבוקרים תהליכי ייצור שלמים, כגון, מתקנים המורכבים ממספר תת-מערכות. הבקרה ברמה זאת היא אדפטיבית בהתאם לתנאי התפעול הדרושים; תדירות פעולות הבקרה קטנה ומורכבות החישובים היא גדולה [3,2].

המחשבים מיושמים כיום בתעשיות התהליכים (process industries), דוגמת תעשיות הכימיות, המזון, החשמל וכו' ובתעשיות הייצור, דוגמאת תעשיות המתכת והאלקטרוניקה, בהם מיוצרים בעיקר פריטים בודדים (דיסקרטיים). בעבר ואף כיום מבוצעות מרבית פעולות הבקרה בתעשיות התהליכים ברמה הראשונה, וכל משתנה מבוקר בפני עצמו באמצעות בקרים אנאלוגיים פשוטים מסוג PID, הניתנים לרכישה כמוצרים סטנדרטיים מיצרנים רבים של מכשור תעשייתי. בתעשיות הייצור מרבית המערכות הן מערכות פיקוד, והן ממומשות באמצעות מערכות ממסרים חשמליים או פנאומטיים ומערכות אלקטרוניות. כל אחת מהמערכות הללו מפותחת ונבנית במיוחד עבור המתקן המסוים מהרכיבים הבסיסיים. בשני סוגי התעשיות מבוצעות בחלק מהמפעלים פעולות פקוח בחדרי בקרה מרכזיים. איסוף הנתונים לחדרי הבקרה נעשה בעבר באמצעות כבלים רב-גידיים, כאשר כל אחד מהמדידים או ממכשירי הבקרה מקושר באמצעות ה"כבל שלו" למערכת המרכזית. שיטה זאת היא יקרה, והוצאות החיווט החשמלי או הפנאומטי הן גבוהות ביותר.

מערכות בקרה הממוחשבות הראשונות שמשו, בעיקר, לפעולות בקרה והשגחה ברמה השניה והשלישית (supervisory control). תוצאות המדידות שבוצעו בתהליכים נמסרו למחשבים בהם חושבו נקודות עבודה אופטימאליות של המשתנים הבודדים. מידע זה נמסר לבקרים האנאלוגיים הסטנדרטיים בתעשיות התהליכים, או לבקרים מוקדשים למשימות מיוחדות, בתעשיות הייצור, ובקרים אלו ביצעו את הבקרה למעשה. המחשבים בתקופה זו היו יקרים ואיטיים מדי ולא אמינים מספיק, ולכן לא התאימו לביצוע פעולות בקרה ברמה הראשונה החייבות להתבצע בתדירות ובאמינות גבוהה. יתרה מזו, כתוצאה מהמחיר הגבוה של מערכות המחשבים לא היה כל יתרון כלכלי בישום מחשבים ברמה הראשונה כתחליף למערכות הקונבנציונאליות. ב-1962 נעשו הנסיונות הראשונים בפיתוח מערכות בקרה ספרתית ישירה (Direct Digital Control)[4]. במערכות אלו נמסרו תוצאות המדידות שבוצעו בתהליכים בתדירות גבוהה ישירות למחשב, כמובן, באמצעות מתמרים מתאימים. במחשב חושב עבור כל משתנה, באמצעות אלגוריתמים מיוחדים, גודלו של אות הבקרה הדרוש, ואות זה נמסר, במישרין בתדירות גבוהה לאלמנט הבקרה הסופי כגון, שסתום פנאומטי או מנוע צעד. הערכות שנעשו באותה תקופה הראו, שכדאי מבחינה כלכלית להחליף את מערכות הבקרה האנאלוגיות הקונבנציונאליות במחשבים שיבצעו את אותה הפעולה, כאשר מספר הבקרים האנאלוגיים המוחלפים היה כ-100. במערכות בהם יושמה הבקרה הספרתית הישירה הוקדש בדרך כלל מחשב אחד לחישובים הרבים והמהירים, שנדרשו ברמה זאת, ומחשב שני שימש למטרות בקרה פיקוחית ברמה השניה והשלישית. לכאורה, נראה היה באמצע שנות השישים, שמערכות בקרה ממוחשבות תחדורנה לתעשיה בקצב מהיר, ושהן תשמשנה לביצוע התיפקודים שתוארו. הציוד הדרוש היה קיים, וגם התשתית התיארוטית היתה מובנה. בהתאם לכך הוכנו גם תחזיות, שהתאמתו, בדבר סוגי ציוד נוספים שיש לפתחם ומחירו הצפוי של ציוד זה[5]. אך למעשה, חלה בשנים שלאחר מכן האטה רצינית ביותר בקצב היישום של מחשבים במערכות פיקוד ובקרה בתעשיה. הסיבות לכך היו רבות:

א. מרבים מפעלי התעשיה לא היו מוכנים לרכוש מערכות בקרה ספרתית ישירה בהן מרוכזות כל פעולות הבקרה האוטומטית של המפעל במחשב יחיד. כל תקלה במחשב זה עשויה לשתק את המפעל כולו. לכן נדרשו מערכות גיבוי אנאלוגיות לבקרת רוב המשתנים במפעל. התקנת מערכות גיבוי אלו ייקרה את המערכות ללא שיעור וללא קבלת תמורה כלכלית או טכנית.

ב. העברת המידע לחדרי הבקרה המרכזיים באמצעות קווים רב-גידיים היתה יקרה ביותר, וכן היו יקרות המערכות הדרושות לתמרת האותות האנאלוגיים לאותות ספרתיים.

ג. פיתוח המערכות ברמה השניה והשלישית חייב פיתוח מודלים מתמטיים מורכבים של התהליכים המבוקרים. המודלים אמנם פותחו, אך חסרו כלים מתאימים לפישוט מערכות המשואות המורכבות שפותחו ולניצול יעיל של המודלים למטרות פיקוד בקרה ואופטימיזציה.

ד. פיתוח התוכנה עבור המערכות היה יקר, וכל מערכת תוכנתה במיוחד, ללא כל תוכנת עזר סטנדרטית למטרות אלו.

<u>מגמות צפויות בשילוב מחשבים בתעשיה</u>

פיתוח טכנולוגיות ה- LSI והמעגלים ההיברידיים, בשנים האחרונות, אפשר להתגבר על חלק ניכר של המגבלות שצויינו. כך ניתנה דחיפה לפיתוח שיטות חדשות בתכנון מערכות בקרה תעשייתיות שיפורטו בהמשך. פיתוח הטכנולוגיות אפשר:

א. ייצור מחשבים קטנים וזולים. לצורך דיון זה אין הבדל בין מחשבי מיני לבין מחשבי מיקרו. שניהם עשויים לבצע תיפקודי פיקוד ובקרה דומים בכל אחת מהרמות השונות. מובן שבכל מקרה ייבחר הרכיב המתאים למטרה.

ב. פיתוח ציוד ריבוב (multiplexing) אמין מאפשר לשדר אותות חשמליים רבים ממדידים רבים או מתת-מערכות המפוזרות בשטח המפעל על פני קו בודד למערכת בקרה מרכזית, ובכך להוזיל במידה ניכרת את הוצאות החיווט החשמלי [6].

ג. פיתוח רכיבים זולים להתמרת אותות אנאלוגיים לספרתיים ואותות ספרתיים לאנאלוגיים. [7]

ד. פיתוח מצגים מתוחכמים, המאפשרים תכנון יעיל של לוחות תפעול מרכזיים עבור התהליכים המפוקחים והמבוקרים.

קושי ניכר בפיתוח מערכות בקרה חדישות מהווה איטיות הפיתוח של גששים (sensors) למדידה רציפה של משתנים רבים בתהליכים התעשייתים. אך גם בתחום זה יש לצפות לשילוב מתמרים, מעגלים משולבים, ומיקרו מחשבים בתוך מדידים ומכשירי מדידה מתוחכמים. זה יאפשר להרחיב את סוגי המשתנים הניתנים למדידה "ישירה" בשדה [8].

כתוצאה מפיתוחו של הציוד שתואר התאפשרה הקדשת מחשבים לביצוע משימות בקרה עצמאיות (dedicated computer systems) [9] ולפתוח מערכות בקרה ממוחשבות מפולגות (distributed computer control systems), כאשר המגמה היא לשלב את המיקרו-מחשבים בתוך התהליך עצמו ולבצע במקום ההתרחשות את מרבית פעולות הפיקוח והבקרה ואת החישובים האפשריים בשיטת הבקרה הספרתית הישירה.

מערכות מקומיות אלו משתלבות במערכות בקרה היראכיות; הן מקבלות הוראות ממחשבים ברמה השניה והשלישית ומעבירות להן מידע על ההתרחשויות בשטח. המידע המועבר לרמה הגבוהה יותר כולל, לרוב, רק מידע על תופעות חריגות שהתרחשו בשטח, המחייבות טיפול של מערכות הבקרה ברמות הגבוהות (management by exception). העברת המידע החריג בלבד מאפשרת גם הקטנה מירבית של כמות המידע המשודר בקוים בין המערכות ברמות השונות. בכך ניתן לפשט את שיטות העברת המידע ולנצל ביעילות יתירה את ערוצי השידור הקוויים, ולעיתים קרובות, כיום, גם את ערוצי השידור האלחוטיים (הצפופים ממילא). לביצוע פעולות בקרה ספרתית ישירה באמצעות מיקרו-מחשבים, הממוקמים בכל אחת מתת המערכות, יתרון גדול לעומת המערכות המרכזיות שפותחו בשנות השישים. כל תת-מערכת במפעל מבוקרת באופן עצמאי, ובמקרה של תקלה במערכת המחשב המרכזית יכולות מערכות הבקרה המקומיות להמשיך בפעולתן השוטפת בשטח, אמנם, בלי דיווח לרמה הגבוהה או קבלת הוראה ממנה. גם במקרה של תקלה באחת המערכות המקומיות הנזק איננו גדול מזה הצפוי במקרה של תקלה במערכות מקומיות קונבנציונאליות.

מרבית מערכות הבקרה הממוחשבות והמפולגות המתוכננות כיום הן מערכות היראכיות, בהן לא קיים, בדרך כלל, כל קשר מקבילי ישיר בין תת-המערכות באותה רמה. הסיבה העיקרית לכך היא חוסר הידע היישומי בתכנון מערכות מחשבים מקבילות למטרות אלו. בתכנון המערכות ההיראכיות קיימת הקבלה רבה לתכנון מערכות עיבוד נתונים מפולגות, כאשר קנה המידה, כמובן, הוא קטן יותר, בהתחשב בגודל המחשבים המפוזרים בשטח המפעל. לכן ניתן יהא להתאים את הידע שפותח במערכות עיבוד הנתונים המפולגות לבעיות שתתעוררנה בתכנון מערכות הבקרה התעשייתיות הממוחשבות. בנוסף לפיתוח הרכיבים הבסיסיים, מחייב יישום השיטות שתוארו גם פיתוח ציוד מודולרי מתאים, שיאפשר את בנית המערכות מתת-יחידות אלה, שתנתנה לרכישה כיחידות סטנדרטיות מיצרני הציוד. זה יאפשר את תכנון המערכות במפעלים והרחבתם בהתאם לצרכים המתפתחים במפעל. הציוד המודולרי שייוצר יהא משלושה סוגים: ציוד שדה לבקרת תת-המערכות, ציוד קישור ודיבוב וציוד פיקוח במרכזי הבקרה. ציוד הפיקוח כולל את המחשבים המרכזיים ואת לוחות המפעיל (operator control console) לוחות מפעיל סטנדרטיים מאופיינים כיום על ידי יצרנים רבים והם ישווקו בעתיד. ציוד המחשבים וציוד השידור והדיבוב כבר סטנדרטי אצל מרבית היצרנים. אמנם, ציודו של יצרן אחד איננו תואם את ציודו של מתחרהו, עובדה המקשה על הקישור בין הרכיבים ותת המערכות של היצרנים השונים. ציוד השדה הממוחשב יהא משני סוגים עיקריים: ציוד סטנדרטי הניתן להתאמה ליישומים השונים, וציוד שיפותח במיוחד למטרות מוגדרות. הציוד הסטנדרטי בתעשיות התהליכים יכלול מרכזי בקרה שונים ל-8 עד 16 משתנים.

כל אחד מהמשתנים יהא ניתן לבקרה באמצעות אלגוריתמי בקרה ספרתית ישירה, מתקדמים ביותר, המתוכנתנים בתוך המערכת והניתנים לבחירה בשדה. מערכות ראשוניות מסוג זה כבר מיוצרות [10,11]. הציוד הסטנדרטי בתעשיות הייצור יכלול מערכות בקרה ספרתית ממוחשבות (computerized numerical control-CNC) הניתנות להתאמה למכונות עיבוד שבבי שונות במפעלים. בנוסף, פותחו כבר בקרים מתוכנתים, [12] (programmable controllers) הכוללים מיקרו-מחשבים, הניתנים לתכנות קל ב"שפה עלית" בשדה. בקרים אלו מהוים תחליף למערכות הממסרים שנבנו בעבר, במיוחד, עבור כל מתקן. ציוד השדה הממוחשב הלא סטנדרטי יתוכנן בהתאם לצרכים, וכמובן שקשה לאפיין אותו במסגרת דיון זה.

פיתוח הגישה המודולרית מחייב סטנדרטיזציה בשני תחומים חשובים: בתחום שיטות הקישור בין ציוד השדה ומרכזי ההפעלה ובתחום התוכנה. בתחום שיטות הקישור אומצו לאחרונה שני תקנים: התקן של CAMAC והתקן בשם Digital Interface for Programmable Instrumentation שהוכן על ידי IEEE . שני התקנים יישמו במערכות בקרה תעשיתיות [13]. בתחום התוכנה יעשה מאמץ לפיתוח שפות עליות, הניתנות לשימוש בנקל על ידי צרכנים פחות מתוחכמים. שפות כאלו כבר פותחו למטרות בקרת ניסויים ותהליכים, לדוגמא: Real Time Basic [14]. מאמצים רבים נעשים לפיתוח תוכנה עלית תקנית במסגרת International Purdue Workshop on Industrial Computer systems

פיתוח השפות העליות חשוב במיוחד כיוון שהצרכנים הביעו בצורה חד-משמעית את אי נכונותם לחזור ולתכנת מערכות בשפות האסמבלר למיניהן [15]. בכתיבת התוכנה יושם גם דגש על שימוש מירבי בשיטות של structured programming . מסתבר גם שכתיבת התוכנה עבור מערכות מפולגות קלה יותר מאשר עבור המערכות המרוכזות. ההתפתחויות השונות בציוד ובתכנון המערכות ישולבו בהתפתחויות חדשות בתורת הבקרה, כאשר יושם דגש על שיטות קירוביות ועל הפרדת בעיות הבקרה המורכבות לבעיות משנה פשוטות יותר [16].
למרות כל הסטנדרטיזציה הצפויה בפיתוח מערכות הבקרה הממוחשבות ידרשו ידע ומומחיות רבים בתחומים התעשייתיים, עבורם יתוכננו מערכות הבקרה. יש לזכור, שתכנון מערכות הבקרה התעשיתיות הממוחשבות מחייב רכישת ידע ומומחיות, לא רק בתורת הבקרה, האלקטרוניקה והמחשבים, אלא גם בתחום המיוחד בו תיושמנה המערכות. כאשר פותחו המערכות הממוחשבות הראשונות היו היצרנים מוכנים לתכנן מערכות עבור כל יישום בכל תחום. מערכות אלו לא היו סטנדרטיות, ומתכנני מערכות הבקרה הפסידו סכומים ניכרים, שהוקדשו מדי פעם ללימוד תהליך חדש או מערכת חדשה. לכן מתמחים כיום רוב המפעלים בחו"ל בייצור מערכות בקרה עבור תחומים תעשייתיים צרים, כגון, בקרת תחנת כוח, מכונות נייר, מכונות דפוס, מתקני בדיקה וכו'.

מערכות ויישומים טיפוסיים שפותחו לאחרונה בחו"ל

תחום היישומים של מערכות בקרה תעשייתיות ממוחשבות הוא, כפי שכבר צוין, רחב ביותר. חלק ניכר מהיישומים החדשים המבוססים על מיקרו-מחשבים טרם פורסם, היות ובפיתוחם הוחל רק לפני כשנתיים, ורק עתה משווקות מערכות חדישות אלו. לכן נתיחס בסקירה זו רק למספר מצומצם של יישומים, המשווקים כבר כיום בהצלחה, ואשר תכנונם איננו מבוסס בהכרח על המגמות שתוארו בעמודים הקודמים. אחד התחומים הבולטים בו יושמו בשנים האחרונות מערכות ממוחשבות היה בדיקות סופיות של מוצרים. הן מנוצלות לדוגמא: בבדיקה אוטומטית של מעגלים אלקטרוניים משולבים[17], בבדיקה סופית של מאיידים למכוניות וכיונונם, בבדיקת מזגני אויר ומקררים, בבדיקת מכשירי קשר ובבדיקת התעיפות של מטוסים. תכנון מערכות הבדיקה המיוחדות הוא קל יחסית כיום, הודות לציוד המדידה, שפותח והניתן לחיבור ישיר ותקני למחשבים, כמו כן הודות לשפות התכנות העליות המיוחדות שפותחו למטרות בקרת ניסויים ובדיקת מוצרים. בתחום תעשיות הייצור משווקות, כיום, בין השאר: מערכות בקרה ספרתיות ממוחשבות (CNC) מתקדמות יותר, מכונות סדר ממוחשבות בתעשית הדפוס וגם מערכות ממוחשבות לבקרת מכונות אריזה לעתונים (מכונה חדישה שדוגמתה כבר הופעלה בישראל).

בתחום תעשיות התהליכים מוקדשים, בנוסף ליישומים המקובלים שנים רבות, מאמצים מחודשים לפיתוח מערכות עבור תעשיות המזון, התרופות והטכסטיל[18], תחומים אשר פיגרו בעבר בניצול מערכות בקרה מתקדמות. בנוסף, מופיעות כבר במספר מפעלים גדולים מערכות בקרת אנרגיה, המווסתות את ניצול האנרגיה במפעלים והמונעות עומסים חריגים (עבורם יש לשלם, בדרך כלל, קנסות גדולים לחברות החשמל). לבסוף, יש לציין עוד יישום טיפוסי חדש והוא, שילוב מיקרו מחשב בתוך מתקן קטן דוגמאת שסתום תעשייתי[19].

יישומים ישראליים והמאמצים הדרושים לשילוב נרחב יותר של המערכות בארץ

בשנים האחרונות יושמו מחשבים במפעלי התעשיה בישראל בעיקר למטרות בדיקות סופיות של מוצרים. הם יושמו במרבית מפעלי האלקטרוניקה לבדיקת רכיבים ומוצרים, ובמספר מפעלים לבדיקה סופית של מוצרים מכניים. בתחום תעשיות הייצור מיוצרות בארץ, כיום, מספר מערכות בקרה ממוחשבות של מכונות כלים. בתעשיות התהליכים יושמו מערכות ממוחשבות שפותחו בחו"ל לבקרת מכונות ייצור נייר ולבקרה של הכנת התערובת במפעל לייצור מלט. בארץ פותחו, נבנו והופעלו בהצלחה מספר מערכות בקרה ממוחשבות, ביניהן כלולות מערכות שקילה במכוני תערובת ומערכות השקיה. מערכות ההשקיה הממוחשבות שפותחו בישראל הן ראשוניות מסוגן בעולם. כן פותחו בארץ מערכות לבקרת מכונות סריגה ומערכות ממוחשבות להכנת הדוגמאות להדפסת בדים.

ההצלחות הללו מורות על האפשרויות הקיימות בארץ בפיתוח מערכות בקרה ממוחשבות בתחומי ייצור בהם אנו מתמחים, כאשר מערכות אילו ניתנות גם לייצוא. אך ההצלחות שפורטו הן נחלתם של בודדים בתעשיה הישראלית. במרבית המפעלים טרם חדרה ההכרה, שמערכות מסוג זה יכולות לתרום רבות לייעול הייצור. הקשיים כאן הם יותר אנושיים מאשר טכנולוגיים. מרבית הצרכנים הפוטנציאליים אינם יודעים להעריך את התרומה הצפויה, את הצרכים ואת האפשרויות הקיימות. הם כלל וכלל אינם יודעים כיצד לאפיין את הפרוייקטים, וכתוצאה מחוסר הידע הם נרתעים מהוסבלים, לאחר מגעים ראשונים עם הספקים. מצד שני, הספקים מעונינים, בדרך כלל, למכור את הציוד המיוחד, אותו הם מיצרים או מיצגים, למרות שבמקרים רבים הוא אינו עונה על דרישות הלקוח. לפתרון מערכת יחסים זו היו יכולים לעזור יועצים, המומחים בתכנון מערכות בקרה תעשייתיות ממוחשבות והמנוסים גם בבניתן ובהפעלתן. הם היו יכולים לעזור לצרכנים באפיון המערכות הדרושות, בתכנונן ובקשר בין הצרכנים לספקים והיצרנים. לפי מיטב ידיעתי, לא פועלים יועצים מנוסים בתחום זה בישראל. יתרה מזאת, תכנון מערכות בקרה תעשייתיות ממוחשבות הוא, עדיין, נושא שולי במוסדות הטכנולוגיים וכמעט שלא נערכים מחקרים יישומים במוסדות אילו בתחום הנדון. יש להדגיש, שבפיתוח פרוייקטים בתחום זה יש משקל רב לנסיון הקבוצה המתכננת והמפתחת. מחיר פרוייקט עשוי להיות גדול פי 4 וזמן הפיתוח יארך, גם כן, פי 4, כאשר צוות הפיתוח אינו מנוסה [20].

לאור מצב זה יש מקום לנקוט מספר פעולות:

א. יש ללמד ביתר הרחבה את הצדדים המעשיים והעיוניים, הכרוכים בתכנון מערכות בקרה תעשייתיות חדישות. לימודים אילו חייבים להתנהל במסגרת קורסים ומוסדות טכנולוגיים בכל הרמות, החל בטכנאים וגמור במהנדסים.

ב. יש מקום להקמת משרדי ייעוץ בתחום זה.

ג. יש לפתח מערכות בקרה ממוחשבות נוספות בישראל בתחומים התעשייתיים, בהם יש לנו נסיון ומומחיות רבה. יש מקום לפתח מערכות אילו, רק אם נוכל לחדש או לייצר מערכות טובות מאלו המיוצרות בחו"ל, כך שיהא ניתן גם ליצאם. היוזמה לפיתוח המערכות הישראליות חייבת לבוא בעיקר מחוגי הצרכנים בתעשיה, כי יצרני המערכות אינם יודעים אילו מערכות דרושות בתעשיות השונות.

ד. יש מקום להקמת קבוצת עבודה של אנשי מקצוע, במסגרת אחד הארגונים המיקצועיים, בה תרוכז פעילות מיקצועית ותוחלפנה דעות באופן שוטף.

סיכום

פיתוח המיקרו-מחשבים וטכנולוגיות ה- LSI מאפשר כיום פיתוח מערכות בקרה ממוחשבות מודולריות וסטנדרטיות, המוקדשות לחלקי מכונות ותהליכים. מערכות אלו תורכבנה בעתיד בכל המכונות ובכל תהליכי הייצור במפעלים והן תופעלנה בצורה היררכית. פיתוח מערכות אלו יזורז על ידי פיתוחים נוספים בתחום הציוד ועל ידי פיתוח שיטות עיוניות נוספות. 40% של רכיבי מחשבי המיקרו שייוצרו בשנים הקרובות יורכבו במערכות בקרה תעשיתיות. היקף המכירות של רכיבי מיקרו-מחשבים שישווקו למטרות אלו בלבד ב-1980 נאמד בכחצי מיליארד דולר לשנה [21].

בארץ כבר מיושמות בהצלחה מספר מערכות בקרה ממוחשבות. יש מקום לפתח בישראל מערכות חדישות נוספות הניתנות גם לייצוא, כפי שכבר הוכח בתחומי תעשיות הטכסטיל והמערכות הרפואיות. הצלחת הפעילות בתחום זה בארץ תלויה בהרחבת מאגר המומחים היודעים לתכנן את המערכות ולהפעילן בצורה תקינה, ובהרחבת הידע של הצרכנים, החייבים ליזום את פיתוח המוצרים החדשים והחדשניים.

רשימת ספרות

1. Eckman, D.P., Lefkowitz, I.,"A Report on Optimizing Control of a Chemical Process", Control Engineering, September 1957.
2. Lefkowitz, I., "Multilevel Approach to Control Syetem Design", 1965 Joint Automatic Control Conference Proceedings, Rensselaer Polytechnic Institute, Troy, New York, June 1965.
3. Mouly, R.J., "Control Trends in Industrial Systems", 21st Annual Instrument Society of America Conference Proceeding, New York, October 1966.
4. Williams, T.J., Ryan, F.M., Progress in Direct Digital Control, Instrument Society of America, Pittsburgh,Pa., 1969.
5. Evers, M.E., Hornor, J.F, et al, "Diversification Opportuniti in Instrumentation", Volume 5, "Process control Instrumentati Stanford Research Institute, Menlo Park, Ca., April 1969.
6. Fling, J., Multiplexing - Once Mistrusted, Now in Demand", Control Engineering, April 1976, pp.87-88.
7. Calkins, R., Berg, A., "Data Acquisition in a DIP Shrinks Systems", Electronics, July 8, 1976, pp. 77-83.
8. Allan, R., "ICs: From Sensors to Displays", IEEE Spectrum April 1976, pp.58-62.

9. Scrupski, S.E., "Processors Moving From Computer Room", Electronics,October 16, 1975, pp. 76-80.

10. (-), "Honeywell Total Distributed Control - TD2000", Honeywell Instrumentation News, No. 74, January 1976.

11. (-), "Low Cost µComputer Controls Wide Range of Industrial Processes", Computer Design, December 1975, p. 106.

12. Andreiev, N., "The PLC Comes of Age - Update Reveals New Markets", Control Engineering, April 1976, pp. 42-44.

13. Linn, E.Y., Schoeffler, J.D., Rose,C.W., "Distributed Microcomputer Data Acquisition", Instrumentation Technology, January 1975, pp. 55-61.

14. (-), "Teach Yourself Real-Time Basic for Measurement and Control", Hewlett Packard Automatic Measurement Division, Sunnyvale, Ca., June 1974.

15. Kompass, E.J., "Control Users Speak Out on Microprocessors", Control Engineering, December 1975, pp. 35 - 37.

16. Peret, R., "Prospects in Automatic Control Research", Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control, December 1974, pp. 376-380.

17. Falk, H., Kaplan, G., "The Microprocessor Takeover", IEEE Spectrum, April 1976, pp. 44-49.

18. Murril, P.W., "The 1985 Control Scene", Instrumentation Technology, January 1976, pp. 39-41.

19. (-), "µProcessor in Control Valve Permits Multiple Use by Software Changes", Computer Design, December 1975, p. 102.

20. Williams, W.D., "Costing Computer Control", Control Engineering, November ,974, pp. 48-50.

21. Baker, W.D., "An Analysis of the Microprocessor Business", Lectur Notes, National Semiconductor Corporation, Santa Clara, Ca., 1976.

אלגוריתמוס לקביעת תואי אופטימלי בפרויקטים-קויים בהנדסה-אזרחית

דוד ערמון

הפקולטה להנדסה אזרחית, הטכניון - מכון טכנולוגי לישראל

1. הצגת הבעיה:

"פרויקט-קוי" הוא פרויקט המשתרע לאורכו של קו (ישר או עקום) בין שני מקומות גיאומטריים נתונים. המקרים האופייניים בהנדסה האזרחית מתיחסים לבנייתם ולהחזקתם של כבישים, מסילות-ברזל, תעלות-מים, קוי צינורות להולכת נוזלים (מים, נפט וכיו"ב) או גאזים, קוי חשמל וטלפון ועוד. בדרך כלל מיועד פרויקט כזה להעברתו של מצרך כל שהוא ממקום אחד למשנהו. תכנונו של פרויקט קוי מתמקד בראש וראשונה בקביעת התואי הגיאומטרי אשר יחבר את שני מקומות המוצא של הפרויקט: "המקור" ו- "הבור". מתבקש לכן פיתוחו של אלגוריתמוס למציאת "תואי אופטימלי", הוא התואי אשר עלותו הכוללת - לטוח של תקופת זמן מוגדרת - תהיה נמוכה יותר מכל אלטרנטיבה אחרת. בעלות כוללת זו משתלבים שלושה מרכיבים עיקריים: עלות הקמה
" תחזוקה
" הפעלה .

כל אחד מרכיבים אלה תלוי מגורמים חיצוניים מגוונים כגון: טופוגרפיה, גיאוגרפיה, גיאולוגיה, אקולוגיה, שיקולים משפטיים, בעלות הקרקע, עלות רכישתה ועוד. מכלול גורמים אלה יכונה להלן בשם "השפעות הסביב".

2. תיאור המודל המתמטי:

א) המסגרת הגיאומטרית: השטח הגיאוגרפי בו חייב התואי לעבור ייחסם בתוך שדה מלבני, אשר יחולק לאחר מכן לסריג של "ריבועים בסיסיים" (ר' תרשים מס. 1). האלגוריתמוס המוצע איננו כופה שום הגבלות על צפיפותו של סריג זה, והמשתמש חפשי לקבוע כרצונו. כקו מנחה יצויינו השיקולים הבאים:

1. השפעות-הסביב יכולות אמנם להשתנות מריבוע אחד למשנהו, אך בתוך כל אחד מהם הן חייבות לקבל ערכים קבועים. מכאן שחייבת להיות קורלציה מסוימת בין צפיפות הסריג מחד לבין המהיימנות של איסוף הנתונים החיצוניים מאידך.
2. הצריכה במשאבי מחשב (זכרון וזמן CPU) עולה במידה ניכרת עם גידול הצפיפות של הסריג.
3. הדיוק התיאורתי של הפיתרון יהיה רב יותר, בה במידה והסריג יהיה צפוף יותר.

ב) השפעות הסביב: עקרונית מאפשר האלגוריתמוס טיפול במספר בלתי מוגבל של גורמי השפעה חיצוניים, כאשר כל אחד מהם מנותח בנפרד. לדוגמא: נניח כי מבחינת התכונות הגיאולוגיות (אופי התשתית הפיסית של הפרויקט) ניתן להגדיר ארבעה סוגי קרקע שונים מהראות של ההשפעה על העלות. אלה יסומנו אז במספרים 4 , 3 , 2 , 1 , ולכל ריבוע בסיסי ייוחס הסוג (והמספר) המתאים.

ג) מרכיבי המערך הכולל של המודל: הפיכתה של הבעיה ההנדסית המוצגת לעיל למודל מתמטי, לגביו ניתן יהיה ליישם טכניקות אופטימיזציה מדעיות, מחייבת ניתוח האספקטים הבאים:

1. איסוף נתונים מקיף ככל האפשר על תכונות השטח הנוגע בדבר.
2. קביעתן הכמותית של השפעות-הסביב לסוגיהן על שלושת מרכיבי העלות.
3. ניסוח מתמטי של המודל ופתרונו.

חקירתם של כל אחד מן ההבטים האלה מהווה משימה נכבדה לשעצמה, ואין המקום כאן להתייחס לכולם. מאמר זה יעסוק לכן רק באספקט השלישי: ניסוחו ופתרונו של המודל המתמטי; והוא יוצא מן ההנחה כי כל המידע ההנדסי והכספי הדרוש למודל רוכז בצורה דטרמיניסטית מושלמת קודם לכן. מאידך יש כמובן לדאוג לתיאום בין המבנה של נתונים אלה לבין דרישות הקלט של המודל.

ד) פונקצית-העלות: יוגדרו הערכים הבאים:

c_1 - עלות ההקמה

c_2 - " התחזוקה

c_3 - " ההפעלה

עלות התחזוקה מחושבת בעלות לשנה, מוכפלת באורך החיים של הפרויקט, ואילו עלות ההפעלה מחושבת כעלות ליחידת מצרך מובל, מוכפלת בסייה הכמות המובלת לתקופת הקיים של הפרויקט. כל שלושת הערכים האלה מתיחסים לאורך הצלע של ריבוע בסיסי, בתואי אופקי ובלתי מופרע. הם עשויים להשתנות מריבוע-בסיסי אחד למשנהו, אך - כמצויין לעיל - הם מבחינת קלט נתון לתהליך האופטימיזציה. בדרך כלל יתבטא קלט זה כשלושה מערכים דו-מימדיים על פני הסריג המוגדר.

ה) השפעות וקטוריאליות: השפעות-הסביב יכולות להיות משני סוגים. ההשפעות המקובלות הן בעלות אופי סקלרי, כלומר הן פונקציה של המקום הגיאוגרפי בלבד. לעומת זאת קיים לפחות גורם השפעה אחד בעל אופי וקטוריאלי, היא מידת השיפוע של התואי (יש השפעה הן לשיפוע לאורך והן לשיפוע לרוחב). בגלל אופיו המיוחד יתיחס האלגוריתמוס אל ההשפעות הוקטוריאליות כחלק אינטגרלי של תהליך הפיתרון עצמו, בו הן תחושבנה בהתאם לצורך הרגעי בלבד. מכאן מתקבל סס"ה העלות לצלע של הריבוע הבסיסי הוא:

$$c = c_1 * q_1 + c_2 * q_2 + c_3 * q_3$$

$q_i \; (i = 1, 2, 3)$ - מקדם המבטא את ההשפעה הוקטוריאלית של הכיבי העלות השונים (פונקציה של השיפועים לאורך ולרוחב של התואי הנבדק).

ו) פרטי המודל המתמטי: עקרונית ניתן לראות את בעית התואי המינימלי כמקרה פרטי או מיוחד של בעית "הנתיב הקצר ביותר" (Minimum Route Problem), המוכרת היטב מתורת-הרשתות. המיוחד בבעיה שלפנינו הוא שאין קיימים בה צמתים וקשתות (Arcs) מוגדרים מראש, ולמעשה המציאות היא של גראף אין-סופי. על מנת שאפשר יהיה להיעזר בטכניקות הידועות של פתרון "הנתיב הקצר ביותר" יש צורך בהכנסתן של שתי מודיפיקציות מתקרבות:

1. הפיכתו של השטח הרציף לגראף סופי, בעל מספר מוגדר של צמתים. משימה זו כבר מצאה את פתרונה עם הגדרתו של הסריג של הריבועים הבסיסיים. תוגדר כל נקודת חיתוך של סריג זה כצומת של רשת. הפשטה זו מחייבת רק כי המקור והבור של התואי יימצאו על נקודות חיתוך כאלו.
2. בחירה של קבוצת קשתות מתאימה, לחיבורם של הצמתים בינם לבין עצמם. זוהי ללא ספק בעיה סבוכה יותר, כי עקרונית הרי ניתן לחבר בקשת כל זוג של צמתים. על מנת להגביל שפע זה של קשתות אפשריות, נבדקו שלושה וריאנטים מציאותיים (ר' שרטוט מס. 2):
   א) המודל של 4 קשתות מכל צומת (מהלכי "צריח")
   ב) " " 8 " " " (מהלכי "צריח" ו- "רץ")
   ג) " " 16 " " " (מהלכי "צריח", "רץ" ו- "פרש").
   ניתוח מתמטי מדויק לחישוב השגיאה המתקבלת כתוצאה מכל אחת מן האלטרנטיבות האלו, נותן את התוצאות הבאות:

מס.	סוג המודל	שגיאה מירבית אפשרית	תוחלת השגיאה
1	"צריח" בלבד	41.42 %	27.32 %
2	"צריח" ו- "רץ"	8.24 %	5.48 %
3	"צריח" , "רץ" ו- "פרש"	2.75 %	1.44 %

מאידך, אין להתעלם מן העובדה כי התשומה הדרושה לביצוע החיפוש של התואי האופטימלי גדלה פי שתיים במעבר מכל אלטרנטיבה אל המתוחכמת יותר.

ז) האלגוריתמוס - תכנות דינמי: עצם האלגוריתמוס לא יתואר כאן על פרטיו, ונסתפק בציון כי המדובר בפיתרון המבוסס על הטכניקה המוכרת של תכנות-דינמי, כאשר ה"שלבים" (Stages) נקבעים דינמית תוך כדי תהליך החיפוש. כל שלב הנו חישוב צומת אשר העלות הכוללת של התואי מן המקור אליו יכול להיקבע סופית. ה"מצבים" (States) של המערכת מתאימים לקשתות השונות באמצעותן ניתן להגיע אל הצומת הבדוק.

3. <u>תואי שאינו צמוד לפני הקרקע:</u>

תכונה אופינית של הבעיה הנחקרת, המיחדת אותה לגבי אלגוריתמים קונבנציונליים של הנתיב הקצר ביותר, קשורה בניתוח האפשרות העקרונית לחבר קטעים של התואי באמצעים שאינם צמודים לפני הקרקע: גשרים (בנייה עילית) או מנהרות (בנייה תת-קרקעית). קטעים כאלה יכונו בשם "קטעים תלושים" (מן הקרקע). כללית ניתן להיקבע כי כל קטע תלוש יחבר שתי נקודות מוצא, כאשר פני הקרקע של כל נקודות הביניים יהיו או במפלס גבוה יותר (מנהרה) או במפלס נמוך יותר (גשר) מן הקו המחבר את נקודות המוצא. למען פשט את החקירה, תיקבענה ההנחות הבאות (שהן מציאותיות ברוב רובן של המקרים המעשיים):

א) הקטע התלוש יהיה קו ישר (לא קו עקום או מתפתל);

ב) הקטע התלוש יהיה מאוזן.

להלן תתנאר שיטת חיפוש המתבססת על שתי הנחות אלו, ואשר באה להשתלב באלגוריתמוס שהוצג לעיל. התיאור יתיחס לנושא של מנהרה (בנייה תת-קרקעית), אך המקרה של קטע תלוש עילי זהה לו בכל פרטיו. על מנת לפשט את ההסבר, הועתקה מערכת הצירים כך שראשה מהלכד כעת (ארעית) עם פתחה של המנהרה:

$$i = (0, 0)$$

האיזור המקווקו בשרטוט מס. 3 יוגדר כ- "הר מקומי" (ביחס לצומת i), הכולל קבוצת צמתים

$$M = \{m_1, m_2, m_3, \ldots\}$$

אם וכאשר מתמלאים שני התנאים הבאים:

א) המפלס h של כל הצמתים $m \in M$ גבוה יותר מזה של הצומת i.

$$h(m) > h(i) \quad \forall m \in M$$

ב) הקו הישר והאפקי היוצא מ- i ומגיע למיקומו של m עובר בתחום שכולו גבוה מהמפלס של i.

תוגדר קבוצה זרה ל- M, היא הקבוצה G של צמתים הגובלים את ההר המקומי M. צומת g יהיה שייך ל- G כאשר:

א) $$h(g) \leq h(i)$$

ב) הקו הישר בין i לבין g נשאר - לכל אורכו - בתחום ההר המקומי.

שיטת החיפוש מתמקדת כעת בזיהויין של כל הצמתים $g \in G$, המתיחסים לצומת של "פתח מנהרה" i נתון. לאחר שזוהתה הקבוצה G המתאימה, תיחשב העלות הכוללת מ- i לכל g, וזאת בהתחשב עם ערכי העלות הספציפיים לתנאי מנהרה.

בדוגמא המתוארת בשרטוט מס. 3 מורכבת הקבוצה G מן הצמתים הבאים:

$$G = \left\{ \begin{array}{l} (0\ ,\ 3)\ ,\ (1\ ,\ 3)\ ,\ (2\ ,\ 3)\ ,\ (2\ ,\ -1)\ , \\ (2\ ,\ -2)\ ,\ (3\ ,\ 1)\ ,\ (3\ ,\ 0),\ (4\ ,\ 4)\ , \\ (4\ ,\ 3) \end{array} \right\}$$

מאידך, לא תיכללנה בקבוצה G נקודות מהסוג של $j = (3\ ,\ 4)$, וזאת מפני שהקו הישר בין i לבין j חותך את קו-הסריג $y = +3$ במפלס שאינו חד-משמעית גבוה מזה של i. במקום מנהרה מ- i ל- j ישקול האלגוריתמוס את האפשרות של מנהרה מ- i לצומת $k = (2\ ,\ 3)$, וקטע רגיל (שאינו תלוש) מ-k ל- j.

בדרך זו תנהג טכניקת החיפוש לגבי כל צומת, כאשר הקו המחבר אותו עם הראשית של המנהרה i אינו חד-משמעית בתחום ההר המקומי M, כגון הצמתים $(4\ ,\ 2)$, $(3\ ,\ -1)$ ועוד.

נשאר כעת להראות כיצד מזוהה הקבוצה G הלכה למעשה. השיטה מתחלקת לשני צעדים:

א) חיפוש לאורכם של שמונה "הכיוונים הראשיים" (ר' שרטוט מס. 4). ייבדק כל כיוון כזה, החל מנקודת הראשית $i = (0\ ,\ 0)$, וכל צומת אשר מפלסו יהיה גבוה מזה של i ישויך לקבוצה M. הצומת הראשון שאינו עונה על דרישה זו (כלומר שגבהו אינו גבוה מזה של i) ישוייך לקבוצה G. בכך מסתיים החיפוש לאורך הכיוון הנידון, והטיפול עובר אל כיוון ראשי אחר.

ב) תוגדר "טבעת" ברדיוס R מסביב לראשית, כקבוצת הצמתים הבאה:

$$\left\{ \begin{array}{ll} |\,x\,| = & R \\ |\,y\,| \le & R \end{array} \right\} \quad \underline{\text{או}} \quad \left\{ \begin{array}{ll} |\,y\,| = & R \\ |\,x\,| \le & R \end{array} \right\}$$

(R - מספר חיובי שלם).

מכאן ש"טבעת" אינה אלא ריבוע בעל צלע $2R$, כשהראשית ממוקמת במרכזו הגיאומטרי.

- בעת ייערך חיפוש לאורכן של הטבעות, כאשר $R = 1\ ,\ 2\ ,\ 3\ ,\ \ldots$ (כלומר, החיפוש מתחיל בטבעת בעלת רדיוס של $R = 1$).

תיבחר כעת צלע כל שהיא של טבעת ברדיוס R , נאמר הצלע y = +R . יימצאו שני צמתים צמודים על הטבעת: (x_i , R) ו- (x_{i+1} , R) , אשר שניהם משתייכים לקבוצה M (כלומר, מוכלים בתוך ההר המקומי). במידה ואכן נמצא צמד כזה, ייבדק הצומת הבא:

$$y = R + 1$$

$$x = \begin{cases} x_i & (|x_i| > |x_{i+1}|) \\ x_{i+1} & (|x_i| < |x_{i+1}|) \end{cases}$$

צומת זה משתייך לטבעת בעלת רדיוס R + 1 . אם המפלס של צומת זה גבוה מזה של הראשית, הוא ישתייך לקבוצה M , אחרת הוא יוגדר כ"צומת-גובלי" וישויך לקבוצה G .

כאמור לעיל, החיפוש יבוצע לגבי כל ארבע הצלעות של הטבעת. עם סיום שלב זה, יעבור החיפוש אל הטבעת הבאה, בעלת רדיוס R + 1 . התהליך כולו יסתיים, כאשר הוא יגיע לטבעת, שאינה מכילה צמתים השייכים לקבוצה M (קצהו של ההר המקומי).

יישומה של הטכניקה יודגם לגבי הדוגמא המתוארת בתרשים מס. 3 :

צעד	פרטים	הקבוצה M x	הקבוצה M y	הקבוצה G x	הקבוצה G y
1	חיפוש לאורך הכיוונים הראשיים (בסדר המצויין בתרשים מס. 4)				
	כיוון מס. 1	0	1		
		0	2		
				0	3
	כיוון מס. 2	1	1		
		2	2		
		3	3		
				4	4
	כיוון מס. 3	1	0		
		2	0		
				3	0
	כיוון מס. 4	1	-1		
				2	-2
	כיוון מס. 5	אין קיים הר מקומי			
	" " 6	" " " "			
	" " 7	" " " "			
	" " 8	" " " "			

צעד	פרטים	הצמד x	הצמד y	הקבוצה M x	הקבוצה M y	הקבוצה G x	הקבוצה G y
2	טבעת ברדיוס 1	(0, 1)	1	1	2		
		1	(-1, 0)			2	-1
		1	(0, 1)	2	1		
	טבעת ברדיוס 2	(0, 1)	2			1	3
		(1, 2)	2			2	3
		2	(0, 1)			3	1
		2	(1, 2)	3	2		
	טבעת ברדיוס 3	3	(2, 3)			4	3
	טבעת ברדיוס 4	סיומו של החיפוש					

4. <u>תכנית המחשב:</u>

הפיתוח כולו בוצע במסגרת עבודת מחקר משותפת בין המחבר לבין Prof. Dr.-Ing. J. Nawrath מן המכון לניהול-בנייה באוניברסיטה הטכנית של מינכן, גרמניה. תכנית מחשב ראשונה נכתבה ב- FORTRAN - IV, והיא משלבת בתוכה את בנייתה של פונקצית העלות (סעיף 2-ד'), ואת עצם האלגוריתמוס של מציאת התואי האופטימלי. בתכנית זו הונח כי קיימת אי-תלות בין השפעות-הסביב השונות, כך שפונקצית העלות ניתנת לחישוב ישיר כחיבור או כמכפלה של הגורמים השונים. הקלט לתכנית זאת הנו:

א) מידע כללי: גודל הסריג, הקואורדינטות של המקור ושל הבור, העלות הבסיסית לצלע של ריבוע-בסיסי (צמוד לקרקע, ובקטע תלוש), המקדמים q של השפעות השיפועים (ר' סעיף 2-ה') בתואי.

ב) המפלס (רום מעל פני-הים) של כל אחת מנקודות-הסריג.

ג) עבור כל גורם של השפעת-הסביב:

1. אופיו (האם יש לחבר את ערכיו אל העלות הבסיסית, או שיש לראות בו מקדם למטרת כפל).
2. מערך דו-מימדי של טיפוסי ההשפעה השונים.
3. המשמעות הכספית של כל אחד מטיפוסי-ההשפעה.

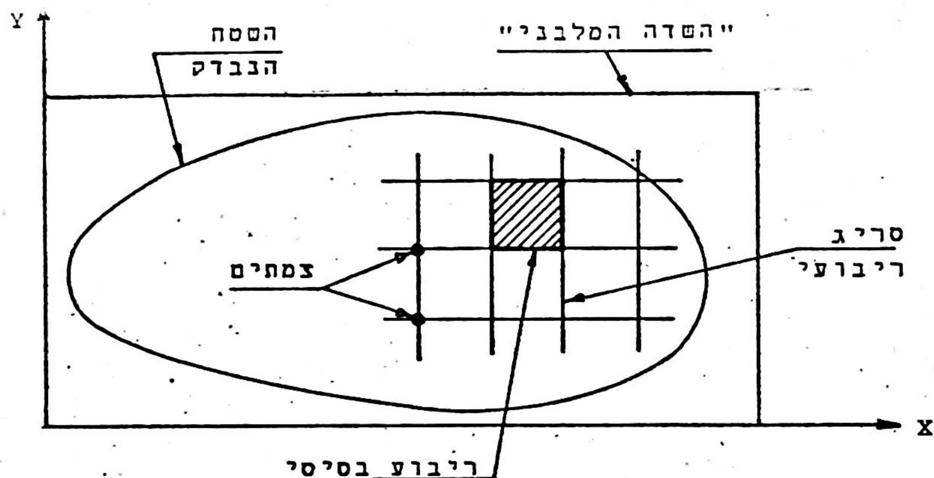
בפלט המצורף ניתן פיתרון של בעיה מציאותית מוגבלת, אשר בוצעה באמצעות תכנית-המחשב הנ"ל. הפרטים הטכניים הם:

הסריג: 55 x 55

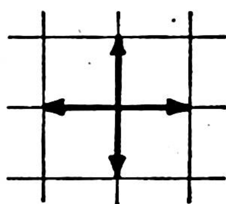
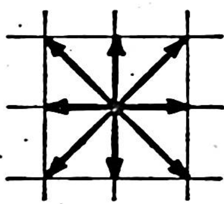
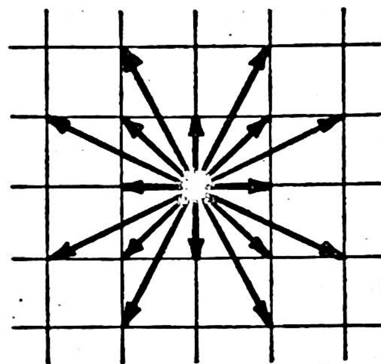
מספר השפעות-הסביב: 2 (האחד מולטיפליקטיבי והשני אדיטיבי).

בבדקו שתי אלטרנטיבות: אין לבדוק אפשרות של מנהרה ושל גשר;
יש לבדוק אפשרות של מנהרה (בעלות כרייה נתון).

זמן המחשב (C.P.U. time) במחשב הטכניון (IBM/370/168) היה כ-20 - 15 שניות לכל אלטרנטיבה.



שרטוט מס' 1: המרכיבים הגאומטריים של המגדל.

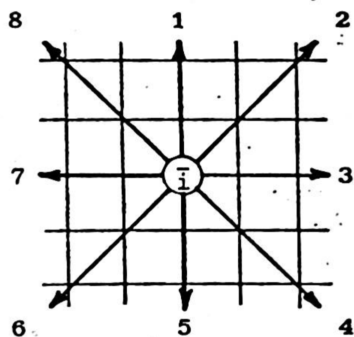




"צריח" בלבד

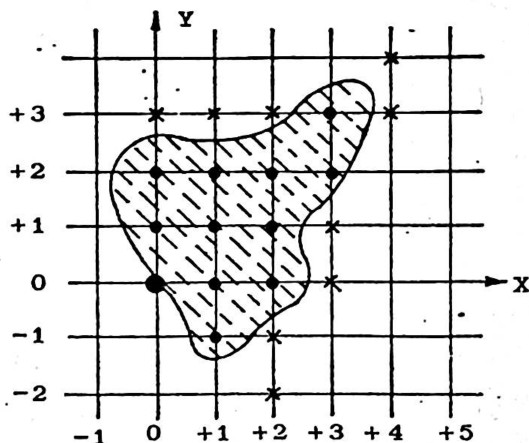
"צריח" ו-"רץ"

"צריח", "רץ" ו-"פרש"

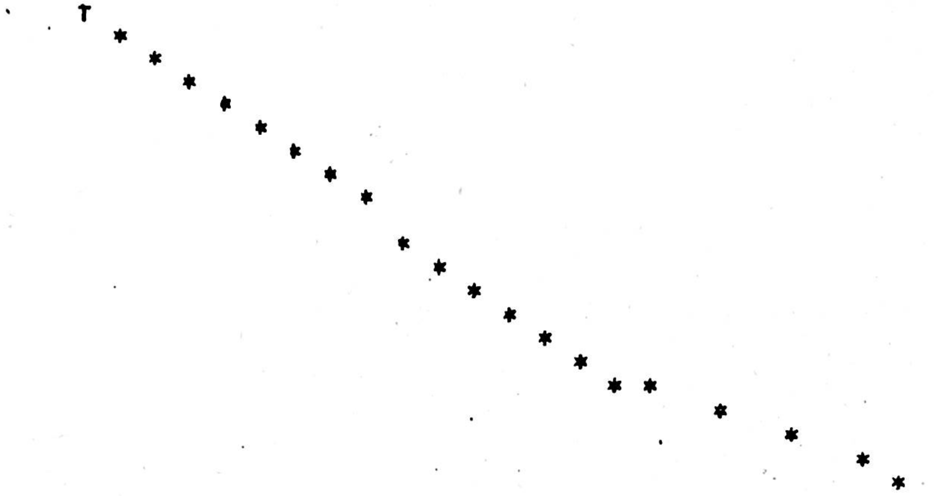
שרטוט מס' 2: אלטרנטיבות למערכת קשתות.



שרטוט מס' 4: כוונים ראשיים.



שרטוט מס' 3: "הר מקומי".

תואי אופטימלי בין שתי נקודות נתונות - כולל מנהרה מתחת לרכס-הרים

תואי אופטימלי בין שתי נקודות נתונות - ללא אפשרות של גשר או מנהרה

A

Z

2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

COORDINATES OF OPTIMUM ROUTE

Y	COST OF STEP	ACCUM. COSTS
46	0	0
46	1019	1019
46	1019	2038
46	1019	3057
46	1023	4080
45	2284	6364
45	1408	7772
45	1038	8810
44	3157	11967
44	1042	13009
43	3148	16157
42	1991	18148
41	6508	24656
41	1454	26110
40	3159	29269
39	2077	31346
38	2394	33740
37	1998	35738
35	2337	38075
33	2332	40407
32	1042	41449
31	1229	42678
29	2498	45176
27	2495	47671
26	1042	48713
25	1042	49755
24	1042	50797
23	1042	51839
22	1042	52881
21	1041	53922
19	2289	56211
17	2288	58499
16	1024	59523
15	1023	60546
14	1023	61569
13	1448	63017
12	1447	64464
11	1447	65911
10	1474	67385
10	1042	68427
9	3168	71595
8	3168	74763
7	2406	77169
6	1554	78723
5	1736	80459

תואי ללא מנהרה וללא גשר

COORDINATES OF OPTIMUM ROUTE

X	Y	COST OF STEP	ACCUM. COSTS
6	46	0	0
8	45	2279	2279
10	44	2283	4562
11	43	1971	6533
12	42	1970	8503
13	40	3106	11609
15	39	3273	14882
24	26	25298	40180
25	25	1604	41784
26	24	1468	43252
27	23	1447	44699
28	22	1447	46146
29	21	1448	47594
30	20	1447	49041
31	19	1447	50488
32	18	1474	51962
33	16	2288	54250
34	15	1474	55724
35	14	1447	57171
36	13	1447	58618
37	12	1448	60066
38	11	1447	61513
39	10	1473	62986
40	10	1042	64028
42	9	3169	67197
44	8	3168	70365
46	7	2406	72771
47	6	1554	74325
48	5	1736	76061

תואי הכולל מנהרה בין הצמתים (15, 39) ו- (24 , 26)

השחלת כבילי עופרת - אופטימיזציה

מאיר כפיר

כבילי עופרת הם אמצעי פיזי להעברת תנועה טלפונית - הם נקראים כך (להבדיל מכבילי פלסטיק) על שם מעטה העופרת העוטף את הכביל ומגן עליו מפני מים והשפעות מזיקות אחרות בקרקע.

משרד התקשורת מוציא מידי שנה עשרות רבות של מליוני לירות לשם רכישת כבילי עופרת (1 מטר כביל עולה מאות לירות). את הכבילים משחילים בין צמתים הנקראים "גובים". המרחק הבין-גובי נע בין 60-90 מטר בדרך כלל. הכבילים עצמם מגיעים על גבי "תופים", כאשר אורך הכביל על תוף נע בין 150-200 מטר.

קיימת, אם כן, בעיה של התאמת חיתוך קטעים מתוך התופים, כך שיתאימו למרחקים הבין-גוביים, והשארית שתיוצר תהיה מיזערית. מרכיב נוסף בפונקצית המטרה הוא מספר המחברים. "מחבר" הוא חיבור פיזי בין שני כבילים ומיקומו - בתוך גוב. כל מחבר מהוה גורם פוטנציאלי לשיבוש הקשר הטלפוני (בעיקר בימי גשמים). גם ההשקעה הכרוכה בביצוע מחבר היא משמעותית. יש ענין רב, אם כן, בצמצום מספר המחברים לאורך מסלול ההשחלה.

כשבדקנו את הנתונים בשדה, התברר כי מידי שנה נמכרת כפסולת כמות ממוצעת של 14% מהצריכה השנתית של כבילי עופרת. הקצאת הכבילים להשחלות נעשתה באופן ידני, כאשר התופים נבחרו באופן אקראי על פי שיקולים של המשחילים או המחסנאים. עם זאת, היתה מגמה בהנהלה להזמין תופים בארכים המתאימים לקטעי ההשחלה של כל פרויקט ולהקפיא מלאי זה עד למימוש הפרויקט. אולם הסתבר לנו ששיטה זו הינה בעלת יעילות מוגבלת, מהסיבות הבאות:

1. שינויים רבים בין התכנון והביצוע, תוך כדי משך זמן האספקה הנמשך כשנה ומעלה. שינויים אלה שמים לאל את הכדאיות בהקפאת מלאי.
2. נזקים הנגרמים לכבילים בדרך מבית החרושת למחסן, גורמים לכך שיש צורך לחתוך את החלק הפגום ואז כל ההתאמה (הכרוכה בתופים רבים נוספים) משתבשת.
3. פיגורים בזמני אספקה של התופים.

עמדה בפנינו, אם כן, הבעיה הבאה:

למצוא שיטת הקצאה שתתן פתרונות תוך זמן קצר (יום - יומיים) ושבעזרתה ניתן יהיה להשיג גם מינימום שאריות של כבילים וגם מספר מחברים מינימלי לכל פרויקט השחלה. לשם כך הומצא אלגוריתם היוריסטי שפרטיו מובאים למטה.

תאור הפתרון

הגדרות

1. שארית - כביל באורך העולה על 30 מטר והנותר כתוצאה מהקצאת חלק מתוף.
2. פסולת - כביל באורך הקטן מ-30 מטר. לכביל באורך כזה אין שימוש והוא נמכר כפסולת נחושת.
3. מסלול - וקטור של מרחקים בין גובים או צירוף של מרחקים כאלה (אם יש אפשרות להשחיל ללא מחבר דרך יותר משני גובים). התנאי שמקיים כל מסלול הכולל את כל הגובים בפרויקט מסוים הוא, שמספר המחברים לארכו הינו מינימלי, האפשרי תחת האילוצים הקימים.
4. קטע - הינו איבר של מסלול והוא מתאפיין בכך שבשני קצותיו יש מחברים.

שלבי התכנית

התכנית מורכבת משני חלקים:

א. הקצאת תופים ללא שארית.

ב. הקצאת תופים עם שארית.

פרוט השלבים

1. סידור מלאי התופים הקיים ברגע נתון במחסן, במרווחים של a מ' בסדר יורד של ארכי התופים.
2. בנית כל המסלולים האפשריים בהתחשב באילוצים של מיקום מחברים הכרחיים ואורך השחלה מקסימלי באותו פרויקט (תלוי בתנאים טופוגרפים ובסוג הכביל).
3. סידור קטעי מסלול במרווחים של a מ' בסדר יורד.
4. סריקות אורכי התופים לעומת אורכי הקטעים, והתאמת זוגות: תוף - קטע. הסריקה מבוצעת b פעמים כאשר ההפרש בין אורך הכביל לאורך הקטע גדל בקפיצות של 1 מ' החל מ-1 מ' ועד ל-b מטרים.

   בסיום שלב זה הוקצו תופים לא מלאים, וכך מנוצלות שאריות הכבילים הקימות באותו רגע במחסן בצורה המקסימלית.
5. מהקטעים שלא נמצאו עבורם תופים בשלב (4) יוצרים את כל הצירופים האפשריים עד לאורך המתאים לאורך התוף המקסימלי שנמצא באותו רגע במלאי.
6. חוזרים על (4) עבור הצירופים שהתקבלו ב-(5).
7. שלבים 3-6 מבוצעים עבור כל מסלול.
8. עבור כל מסלול מחושב יחס הפסולת והמסלול שיחס זה מינימלי עבורו, הוא המסלול הנבחר. בסוף שלב זה הסתיימה הקצאת תופים ללא שארית.
9. חיתוך קטעים להשחלה

   קטעים אשר לא נמצא עבורם, או עבור צרוף שלהם עם קטעים אחרים, כביל באורך מתאים, יחתכו מתוף כך שהשארית שתוותר תהיה בעלת סיכוי מירבי להיות נדרשת בעתיד הקרוב.

כדי להשיג זאת יש לוודא שההסתברות לדרישת האורך שהתקבל תהיה קרובה ככל האפשר לשכיחות היחסית של אורך זה במלאי התופים הקיים.
לא נפרט כאן את פונקצית המטרה המתאימה שפותחה לצורך כך. כדאי לציין רק זאת: הסתברות הביקוש לקטעים באורכים השונים, נבנתה על סמך אינפורמציה היסטורית של אורכי השחלות. אולם התפלגות אורכי הקטעים המגיעים לשלב חיתוך אינה זהה להתפלגות הביקוש, משום שלקטע קצר הסתברות גבוהה יותר שימצא עבורו, או עבור צרוף שלו עם קטע אחר, כביל-בשלב הסריקה.
לכן, התוכנית לומדת תוך כדי הרצתה את התפלגות הקטעים המגיעים לחיתוך ומעדכנת עצמה בהתאם.

<u>הקלט והפלט של התוכנית</u>

<u>הקלט:</u>

1. סוג הכביל.
2. מספרי וסוגי הגובים, והמרחקים ביניהם.
3. פרוט מיקום מחברים הכרחיים.

התכנית מכילה טבלה קבועה שבה רשומים הנתונים הבאים:

1. אורך מקסימלי מותר להשחלה, לפי סוג הכביל והצנרת.
2. סוגי גובים (לפי קודים) והשאריות שיש להוסיף למחבר או לכפוף לכל סוג גוב.

<u>הפלט:</u>

הפלט מיועד עבור שני גורמים:

א. קבוצת ההשחלה.
ב. מחסן הכבלים.

א. <u>פלט לקבוצות השחלה</u>

פלט זה מכיל את תכנית ההשחלה: מספרי הגובים שבהם יש לבצע מחברים, אורכי קטעים, מספרי התופים המיועדים לקטעים ואורכיהם.

דוגמא:

מחסן	מסי תוף	אורך התוף	אורך קטע השחלה	מגוב	דרך גוב ביניים	דרך גוב ביניים	לגוב
סקייה	97	190	118.5	5251	5252	-	5253
			71	4107	-	-	4108

ב. פלס למחסנים

פלס זה מכיל את ההקצבה המיועדת למחסן האזורי: מספרי התופים, ארכיהם וסימון הפרויקט.

דוגמא:

פרויקט	מס' תוף	אורך התוף
איילון	451	180
	453	201
	301	88

תוצאות

בטבלה למטה מובאות תוצאות לדוגמא של השימוש בתוכנית "KVIL". ניתן לראות כי אחוז השאריות הממוצע היה 0.5 לעומת-14 בשיטה הידנית -

מובא גם לצורך השואה מספר המחברים בכל פרויקט כפי שתוכנן ידנית, לעומת ההקצאה ע"י "KVIL".

פרויקט	אורך: הפרויקט	שארית	% שארית	מס' גנבים	מס' מנצע מחברים	מס' קיים מחברים
1	1750.5 מ'	9.5 מ'	0.54	31	16	31
2	1646.5	6.5	0.39	25	17	18
3	1419.0	9.5	0.67	24	15	14
4	1627.5	18.0	1.11	23	16	18
5	981.5	2.0	0.20	14	10	8
6	1871.5	10.0	0.53	28	19	24
7	1463.0	9.5	0.65	25	14	20
8	1770.5	11.5	0.65	29	20	24
9	1709.0	7.5	0.44	26	18	20
10	1614.0	1.5	0.09	33	14	26
				סה"כ	159	203

התפתחויות נוספות

בעיקבות הצלחת "KVIL" בעשרות פרויקטים בת"א ובירושלים, התבקשה המחלקה לחקר באמצעים למצוא תשובה גם לשאלה הבאה:
מתוך ההזמנה השנתית של כבילים (15 ק"מ למשל), כמה תופים יש להזמין מכל אורך?
כדי לענות על שאלה זו הוכנה תוכנית סימולציה אשר בחנה את ההשפעות של הרכבי תופים שונים על יעילות ההקצאה.
התקבלה טבלה דוגמת זו שלמטה:

אורך התוף (במטרים)	מספר התופים להזמנה
140	5
150	20
170	30
180	30
190	40

סוף דבר

התכנית "KVIL" מביאה להקטנה דרסטית (עד לאפס כמעט) בשעור הפסולת הנוצרת בתהליך השחלת כבלי עופרת.
השימוש בתוכנית זו עד היום כבר הביא לחסכונות ניכרים.
הכנסת התוכנית כתוכנית סטנדרטית להשחלות בכל הארץ, תביא לחסכון שנתי של 4-5 מליון לירות (במחירי 1976).

קבוצת דיון מס' 3:

"הערכת ביצועי מחשב"

יו"ר: ל. ספר

ג. קנדלר

א. רונה

נ. למלבאום

AN MBO APPROACH TO CPE

A lot of attention has been given in recent years to methods of Computer Performance Evaluation (CPE). When reviewing the literature [1] one standard performance measurement technique does not appear to have been applied to CPE. This is Management By Objective (MBO). This article outlines a methodology for applying MBO to measuring the performance of an operations department.

1 The Scope of the Operations Department

Figure 1 is a diagram of an operations department that provides a number of user services:

Local Batch
Remote Batch
Online Applications (e.g. IMS DB/DC applications).
Time Sharing (e.g. TSO application development).

In the diagram, and for the purposes of this article, it is assumed that operations are responsible for all local equipment and communications lines only. The terminals themselves are a user responsibility. The duration of the operations department period of responsibility is defined in Table 1.

2 The Function of the Operations Department

The function of the Operations Department is to provide service that is 'needed' or 'requested' by its users. Much work has been done, particularly by Streeter [2,3,4], on providing an objective methodology for evaluating the service needed by an individual user of a DP Centre which services a single enterprise. A commercial data servicer is in business to provide the service requested by its users. In most installations, however, service levels can be agreed between operations and the users which represent a fair compromise between user 'wants' and their 'needs'.[5]
The objectives will be, therefore, to provide agreed levels of service, to system users, in response to their submission for processing of different types of transaction.

3 Defining Service Levels

In MVS the term service rate is used to describe the agregate rate of consumption of three key system resources used during a jobs execution. A more generally applicable definition is

./.

Table 1: Duration of Operations Responsibility for Transaction

Type of User	Type of Transaction	Period of Responsibility From	To
Local Batch	Job See note 1	Receipt of Job Input	Return of Job Output
Remote Batch	Job See note 2	First card image	Last print line transmitted
Online Application	Application Transaction	Last character received	Last character transmitted
Timesharing	Interaction	Last character received.	Keyboard unlocked

Note 1: It is assumed that job assembly and quality control are performed by a separate department.

Note 2: Remote Batch stations may belong to users, as shown here, or may be considered an extension of the central site. In this case the responsibili of the operation department are as described for local batch. Conversely, all unit record equipment - including locally attached units - may belong to users and operations responsibility may be as defined for remote batch. A more generally applicable segregation of user types would be 'Operations Submitted Batch & User Submitted Batch'.

$$\text{Service Rate} = \frac{\text{Agregate Resources Provided}}{\text{Length of Transaction}}$$

A more valuable formula is:

$$\text{Transaction Service Rate Required} = \frac{\text{Maximum Resources Requestable}}{\text{Maximum Length of Transaction.}}$$

Example: A terminal transaction must have a 3 second response time of which not more than 1 second can be line time for the reply. The line is 4800 Baud and the maximum message length is 300 characters. The service rate requred from the line, during this subset of the transaction, is 1/2 second per second. As with MVS we will agregate the service requirement, rather than specify each component as in the example, and the result is a list of services offered. A partial example is given in Table 2.
The rest of this article will concentrate on a batch example.

./.

Table 2: EXAMPLE OF SERVICES OFFERED BY A DATA CENTRE

Service Type	Maximum Resources Requestable (Note 1)	Service Level Offered
Batch: CLASS A	min. CPU, 10,000 lines printed. No mounts.	90 minute turnround
CLASS B	As for Class A with up to 2 tape mounts	120 minute turnround
⋮	⋮	⋮
CLASS E	As for class A - Express turnround	30 minute turnround
⋮	⋮	⋮
CLASS Z	No limits	24 hours turnround
TSO: TRIVIAL	INPUT, INSERT DELETE subcommands of EDIT.	3 second response time
⋮	⋮	⋮
COBOL	Not more than 300 source cards	3 minute response
IMS: Inquiry for balance, name or address.	100 character response	1 second response time
⋮	⋮	⋮
Insert new record.	300 characters entered	3 second response time.

Note 1: Stated in terms of limiting factors the user can understand and control.

4 Reporting

Figure 2 shows a typical report produced for management. The danger of this type of report is best described by Bryan:
"We rarely see the average - we usually see the typically."
Figure 3 represents a step in the right direction. It is obvious from this report that CLASS B users are getting much more consistent service than CLASS A users. In order to develope this example it is important to understand what is meant by turnround.
Figure 4 diagrams the phases of a batch job. All the timestamps within the box marked 'Response to Operator Controlled Workstations' can be, and very often are, recorded by the operating system (e.g. SMF Record Type 26). The job receipt and delivery timestamps must be made manually

./.

(or via a system such as the 5320). As a result, more often than not, they are not made. In many installations this leads to disagreement between operations and the users as to what turnround time actually is.

Having said this, the turnround time for both CLASS A and CLASS B in figures 2 and 3 is from first card read to last line printed. The difference between them is that CLASS A jobs are submitted and printed at the central location, whereas, CLASS B jobs use a private printer/card reader combination in another building attached via a multiplexor repeater. A breakdown of the response components revealed that the big variation for CLASS A users was entirely in the 'Q FOR OUTPUT' element of response. Further investigation showed that CLASS A users shared two printers with CLASS C production jobs. This meant that there were very often 2 CLASS A jobs printing simultaneously but, occassionally, there were 2 CLASS C jobs printing simultaneously to the exclusion of CLASS A jobs. In this case an adjustment to the SYSOUT classes and priorities solved the problem and produced consistently 'acceptable' CLASS A turnround.

Every MBO advocate has now 'seen red'. Acceptable is not a word we use in MBO, and it is obvious that, despite the improvement in reporting that printing a distribution provides, it is still not Management By Objective. Let us restate an objective for CLASS A jobs.

a. Transaction Definition: A CLASS A job may use up to 30 seconds of CPU time and print up to 5000 lines. No setup requests are honoured for CLASS A jobs.

b. Service Objectives: Provide two hour turnround (first card read to last line printed) for all CLASS A jobs.

c. Control Measurement: 80% of jobs should receive 1/2 hour turnround. (With the usually encountered distribution of turnround times, this will ensure that 2 hour turnround is provided except where there is a system failure. It is also, as will be seen, a more sensitive trend measurement).

Figure 5 shows a monthly management report which shows graphically when we have made and when we have missed our targets and by how much. It can be seen that on the 11th and 12th of the month there was a sudden severe degradation in turnround time, the 1/2 hour target being missed 4 times.

We can now introduce another management technique - Exception Reporting. Whenever objectives are being met nothing need to be done. However, if they are not being met, corrective action must be taken. The sooner this action is taken the more effective it is. Thus, a second report should be a daily report similar to that shown in figure 6. The report is a printout with the last column ('reason') left blank. A daily meeting is held at the instigation of the operations manager to assign reasons

to each target that is missed. Where the reason the target was missed is obvious, it is filled in at the meeting. Where it is not, responsibility for investigation is assigned to one person who reports back at a predetermined future meeting (normally the next day). This would be handled as part of a problem tracking system.[7,8] In this way problems outstanding from the daily summary are regularly brought before the DP manager and his management. As can be seen from figure 6, the problem causing missed objectives on January 11/12 th is, what I have called elsewhere [9], an availability problem.

Turning to the end of the month, the most obvious reason for slower turnround is a higher workload. In the case of short jobs this is normally equvalent to an increase in the number of jobs. In the case of longer batch jobs (e.g. a clearing banks overnight work) workload may correlate better with the number of input transactions. A similair correlation would apply between the transactions of a timesharing or DB/DC subsystem and the workload it places on the system.

Figure 7 is a modification to figure 5 which shows this standard measurement on the same chart as turnround is reported. It can be seen here that our 1/2 hour target is much more sensitive to small changes in workload than the 2 hour objective. With standard measurement of this sort we can start our daily investigation with the most obvious source i.e. an increased workload. However, we will not see on any one day's report whether we have an exceptional day or a genuine increase in workload. The monthly report, figure 7, shows that in the last 10 days of the month we missed our 1/2 hour target 4 times and on each occasion the number of CLASS A jobs exceeded 300. The question that will now come to any manager's mind is 'Is this an isolated incident or do I have a trend?' Figure 8 is a plot of 3 key measurement for each of 14 weeks ending on 3rd April.

These 'unsmoothed' graphs are not very useful for showing trends. A superior technique is that of the smoothed graphs shown in figure 9. For observing whether our 2 hour turnround target is being met we take the average of the percentage of jobs acheiving this target in each of the last 4 weeks. If this figure consistently falls below 95% we have a very serious problem in performance or availability.

One pointer to which of these is responsible (in the case of CLASS A jobs but not necessarily production jobs) is the Standard Deviation of turnround. Rather than plot this rather meaningless figure, we plot the 80th and 95th percentiles of turnround time. A marked widening of the gap between them, i.e. an increase in the Standard Deviation, will often point to an availability problem. (In this type of environment, submission of a job is normally dependent on receipt of results from a previous job. Thus, with the system down for a short time - up to two hours - there will not be an inordinate build-up in the number of jobs waiting to be processed. Jobs which were in the system before the breakdown will have very long turnround

./.

times, jobs entering immediately after the breakdown will suffer a small elongation in turnround time but most programmers will 'lose a run'. Thus, the 95th percentile of turnround time will increase more significantly than the 80th percentile. Because there are other possible explanation for the diversion of the 2 lines, this may be used as a first theory for investigation but not taken as conclusive proof).

For plotting the two percentile curves in figure 9 I have used a different smoothing technique. In this case the new average (A) is equal to the old average multiplied by 4 plus this weeks measurement divided by 5. In this way, the measurement for any week is worth 20% of the smoothed average in the week it occurs, 16% the next week, around 5% in the 7th week and just over 1% in the 14th week. This is a very useful tool for observing underlying trends whilst smoothing out 'freak' peaks. From figure 9 it is obvious that there is no significant underlying trend.

Resource Useage Reporting

Up to this point each objective has been viewed in isolation. However, in a large system, there will obviously be interactions between the different types of work. Thus (refer back to the example in 3.1.3) if CLASS E, TSO trivial and IMS inquiry work all increase and we maintain their service levels, we are likely to see a degradation in the service to CLASS A users. To be able to see this we need reports on common denominators between the different types of services. These common denominators are the physical system resources e.g. CPU time used, EXCP's (EXecute Channel Program - a macro that leads to an I/O operation) issued, lines printed, disk space occupied etc.

If we take CPU useage as an example we will first have to divide the day into a number of periods which represent peak and non-peak hours. Table 3 is an example of how each of the different types of workload will vary over a 24 hour day.

For each of these periods we can produce daily and monthly consumption reports as well as trend reports, which break down the available CPU time into that used by TSO, IMS and each of the batch classes, wait time and system time. If we use OS Systems Management Facility (SMF) as our source, we will probably want to break the system time down into that due to the non-recording of CPU time used in connection with I/O activity (EXCP's) and 'other system time'.

Table 3: Workload composition by time of day.

	TSO	Batch Testing	Batch Production	IMS
0800 - 1000	HEAVY	HEAVY	MODERATE	VERY HEAVY
1000 - 1200	VERY HEAVY	VERY HEAVY	LIGHT/MODERATE	VERY HEAVY
1200 - 1400	HEAVY	VERY HEAVY	LIGHT	HEAVY
1400 - 1600	VERY HEAVY	HEAVY	MODERATE	MODERATE
1600 - 1800	MODERATE	HEAVY	MODERATE	NONE
1800 - 2200	LIGHT	LIGHT	HEAVY	NONE
2200 - 0400	NONE	NONE	VERY HEAVY	NONE
0400 - 0700	NONE	NONE	LIGHT/MODERATE	NONE
0700 - 0800	LIGHT	NONE	NONE	HEAVY

Table 4: Summary of Reports Produced

REPORT	CONTENTS & FORMAT	DISTRIBUTION LIST
Monthly Summary	Achievement versus target. Graphical with explanatory notes.	Operations Manager, Shift Leaders, DP Manager, DP Manager's boss, users (See note 1).
Daily Summary	Objectives not met. List for action/ investigation.	Operations Department, Systems Programming, DP Manager (See Note 2).
Trend Reports	Effect of less than 100% availability, Short & long term trends in service. Graphical with explanation.	DP Planning, DP Manager, DP Manager's boss.

Note 1: Many DP Managers want to hide their performance from their users. This is a fatal mistake. If the users have agreed to service levels they are entitled to know whether these are being met. More often than not they will be. When they are not the user will often be a useful ally to the DP Manager in his attempt to justify an increased capital budget to senior management. Senior Management often see

./.

the DP centre as a straight overhead cost without seeing the benefits that accrue to the users as offsetting this. Bringing the user and his requirements into the DP budget cycle will normally result in DP getting a fairer share of the overall budget.

Note 2: This report goes to the DP Manager after the daily meeting and gives him a picture of what problems occured, which have been resolved, which are under investigation and who is doing the investigation. It should be cumulative with outstanding problems - and their date of origin - re-appearing until they are resolved.

6 Distribution of Reports

Table 4 summarizes the reports that must be produced and the distribution list for each.

7 Increasing Capacity

The standard premise of CPE advocates is that the system should always be tuned to the maximum. 'Maximum' is another word that MBO banishes, because, in this context, it is a subjective and unmeasureable term. However, there is no doubt that systems can be tuned and that the effect of this tuning can be very substantial. However, system tuning in OS (and OS/VS) systems, in my experience obeys the Law of Diminishing Returns. There is a substantial list of easy to implement tuning techniques which invariably improve system performance.

After this we rapidly slide down the graph of the diminishing returns curve until we reach the point where the cost of investigating and implementing a performance improvement exceeds the benefit derived.

Conversely, within the IBM hardware CPU line - 115, 125, 135 and, more relevantly to OS customers, 145, 158 and 168 - there are big gaps in size between the different offerings. A 158 is roughly 3 times the power of a 145 and a 168 is over twice the power of a 158. Thus when moving from one CPU to the next the normal situation is that there is initially an excess of computing power.

If a tuned system is operated on a 168 which has replaced a 158 the effect is to double capacity. If we assume that the 158 was tuned to 'the maximum', and was just able to provide all needed service, the effect will be that users will receive better service than they need, but, will come to view this as normal and will develope higher expectations. These expectations will become de facto needs.

It is my suggestion that we conceal part of this power, by operating an untuned system when we initially install, and implement known tuning techniques to maintain service levels. When we have a 'standard' tuned system, which is

./.

operating near capacity, we should order additional hardware to increase system capacity. We can engage in more costly forms of tuning, until that hardware arrives, without ever reaching the point on the diminishing returns curve where we are getting a net negative return on our investment in tuning.

8. Implementing Service Level Reporting

8.1 Setting Objectives

Firstly, DP management should review what services they are, de facto, offering i.e. what types of service can the user request. The next phase is to measure what service levels are being provided. Having done this, DP Management should consult with the users and set an initial set of objectives. These should be publicized with the proviso that they will hold for a trial period during which DP Management will accept requests for modification from users. At the end of the trial period modified objectives are set. The key to a successful implementation is that users be charged premium rates for premium service and can receive a discount if they accept service with poor or non-existent guarantees.

8.2. Data Collection and Data Reduction

CPE enthusiasts tend to be xenophobic about overhead. One of the places in whihc they display this xenophobia is by trying to avoid the overhead of measurement. This may be excellent in a laboratory environment but is unacceptable in business. The cost of measurement on a typical production line is 5 - 10% of total budget and these are valid guidelines for a DP environment.

In an OS or OS/VS system most of the data collection is built into the system. For batch we have SMF, for TSO we have TSTRACE or, in MVS, GTF SYSEVENT Trace. Packages like IMS build their own log tapes. For data reduction the author reccomends the acquisition (purchase, SHARE library, exchange with other installations) of existing packages to produce reports. In addition there is a lot of information in the system log and in manual operator logs that is useful for investigation. Once again, existing programs are available to process this data.

Optionally, some of these traces may be run only at specific times of day or days of the week. Using Table 3, for example, it might be decided to start TSTRACE only between the hours of 1000 and 1200 and then only on Sunday, Tuesday and Thursday when a specific production run moves the batch production load from 'LIGHT' to 'MODERATE'. (A standard TSO throughput measurement - total transactions or transactions per connected minute - can be developed to monitor TSO load at other times via SMF. SMF will in any case monitor TSO user resource consumption).

9 SUMMARY

1. A DP centre is in business to offer service to its users.
2. The services offered should be clearly defined and, externally announced, objectives for level of service should be set. Additional internal objectives, for control purposes, should also be defined.
3. The degree of achievements of these objectives should be measured and reported.
4. System tuning should be used to maintain these predefined levels of service.
5. The trend of achievement of these objectives should be tracked to predict long term needs for additional capacity.

References:

1. For example, EDP Performance Review, Volume 4 Number 3, 'Bibliography of 1975 Performance Literature' and Volume 3 number 3 'Bibliography of 1974 Performance Literature'.

2. Streeter, D.N., 'Productivity of Computer Dependent Workers', IBM Systems Journal, Volume 14 number 3.

3. Streeter, D.N., 'Cost-Benefit Evaluation of Scientific Computing Services', IBM Systems Journal, Volume 11, number 3.

4. Streeter, D.N., 'The Scientific Process and the Computer', Section 3 'Issues of Management', John Wiley and Sons Inc., New York.

5. See: Kleinrock, L., 'Optimum Bribing for Queue Position', Operations Research Volume 15, number 1, for a method by which this approach may be developed.

6. Bryan, G.E., 'JOSS, 20,000 Hours at a Console - A Statistical Summary', Afips Conference Proceedings, 1976 FJCC.

7. 'MCAUTO Hierarchical Communications in DP Management', IBM Form Number GK20-0864.

8. BELHUMEUR R, et al, 'IBM System/370, Tracking and Resolving Problems and Co-ordinating Changes', IBM Form Number GG22-9000.

9. Kandler, G.M., 'What is Availability?' IBM Israel Ltd IM Newsletter number 2 of 1976.

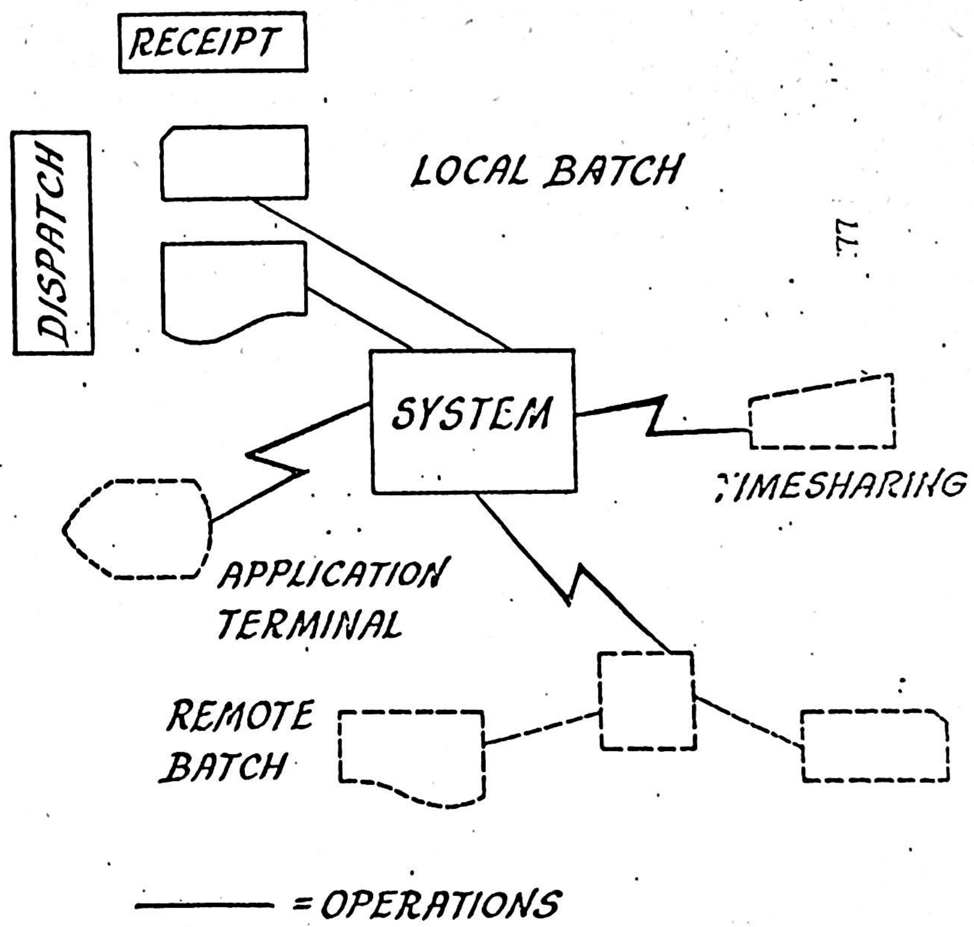


Figure 1 - The Scope of the Operations Department.

CLASS	TOTAL JOBS PROCESSED	AVERAGE EXCPs PER JOB	AVERAGE CPU TIME PER JOB	AVERAGE TURNROUND TIME
			MM. SS.	HH. MM.
A	6273	184	00.03	00.42
B	1472	275	00.12	00.34
C	216	18,162	33.42	06.26
D	875	4,572	06.53	08.51
E	1927	282	00.23	00.59

JOB TURNROUND SUMMARY–
NOVEMBER, 1975

Figure 2 – Typical Management Report on Job Turnround.

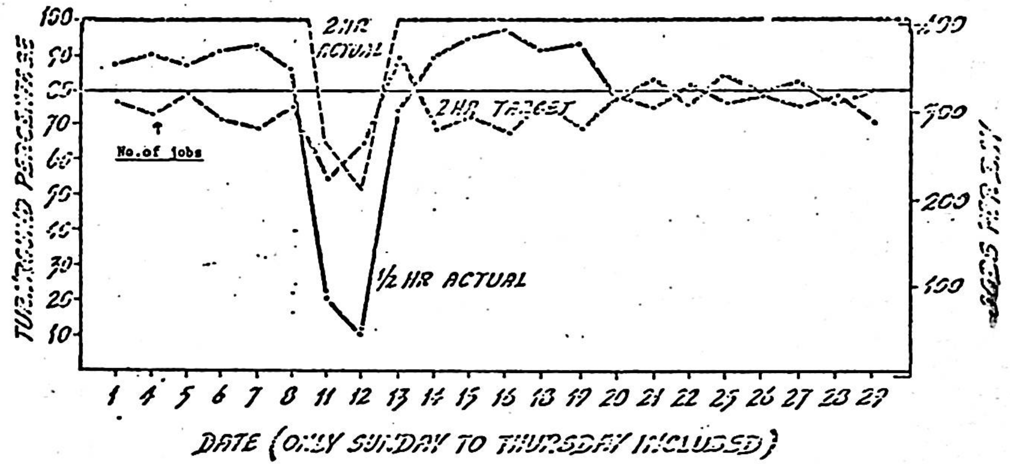
CLASS	TOTAL JOBS PROCESSED	AVERAGE TURNROUND TIME	80%	95%	100%
			COMPLETED IN UNDER OR EQUAL TO:		
		HH.MM	HH.MM	HH.MM	HH.MM
A	6273	OC.42	00.27	01.51	C4.4S
B	1472	00.34	00.36	00.44	02.15
C	216	06.26	05.28	9.49	21.26
D	875	08.51	10.02	18.37	38.11
E	1927	00.59	1.20	2.47	11.42

JOB TURNROUND SUMMARY–
NOVEMBER, 1975

Figure 3 – Facts Hidden by the Average.

Figure 7

MONTHLY REPORT OF SERVICE TO CLASS A USERS
JANUARY, 1976



DAILY SUMMARY

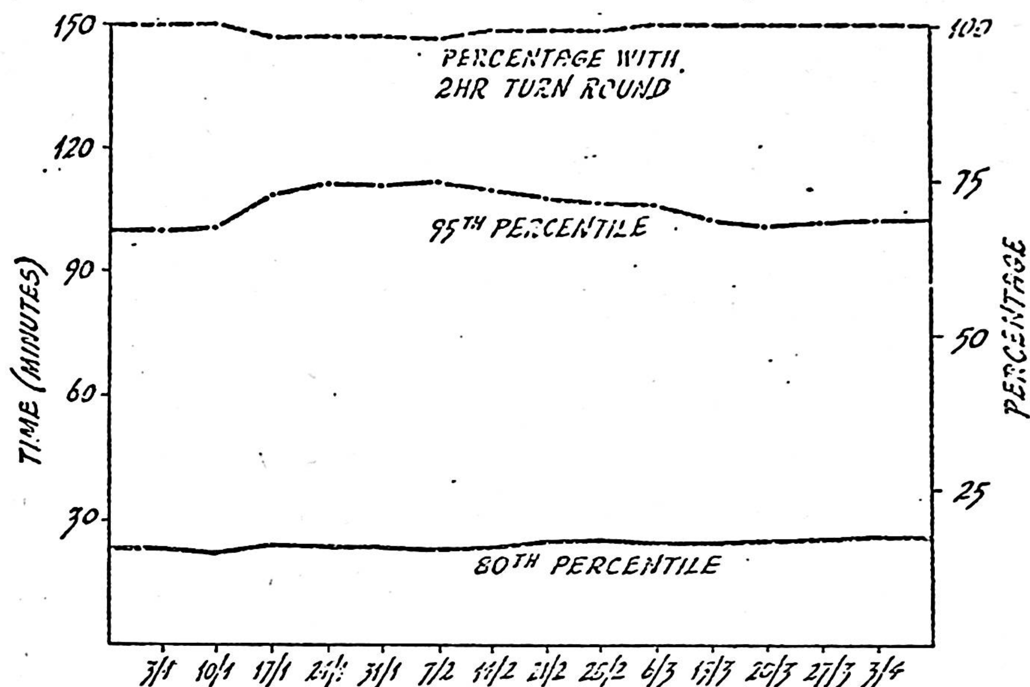
OBJECTIVE	ACTUAL	REASON*
80% OF CLASS A JOBS TURNROUND IN < 1/2 HR	15.3%	ONE SCRATCH PACK NOT OPERATIONAL
100% OF CLASS A JOBS TURNROUND IN < 2 HRS	65%	— " —

SUMMARY OF OBJECTIVES NOT MET 11/1/76

* FILLED IN BY HAND AFTER INVESTIGATION

Figure 6 - Daily Summary Report

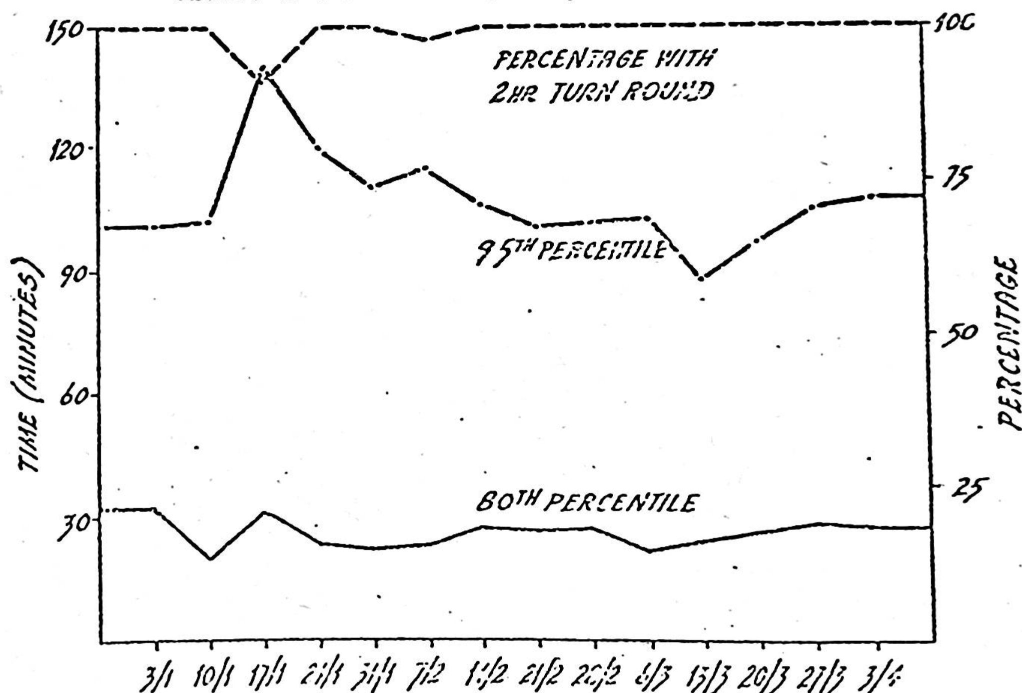
CLASS A TURNROUND ~ SMOOTHED



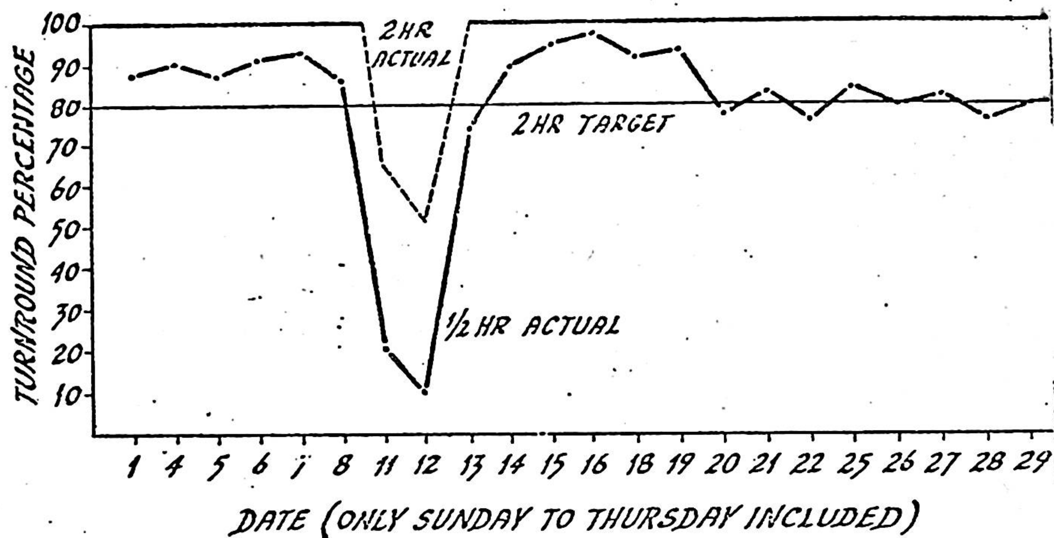
——— —·— $A = \frac{(A \times 4) + \text{THIS WEEK}}{5}$ OR

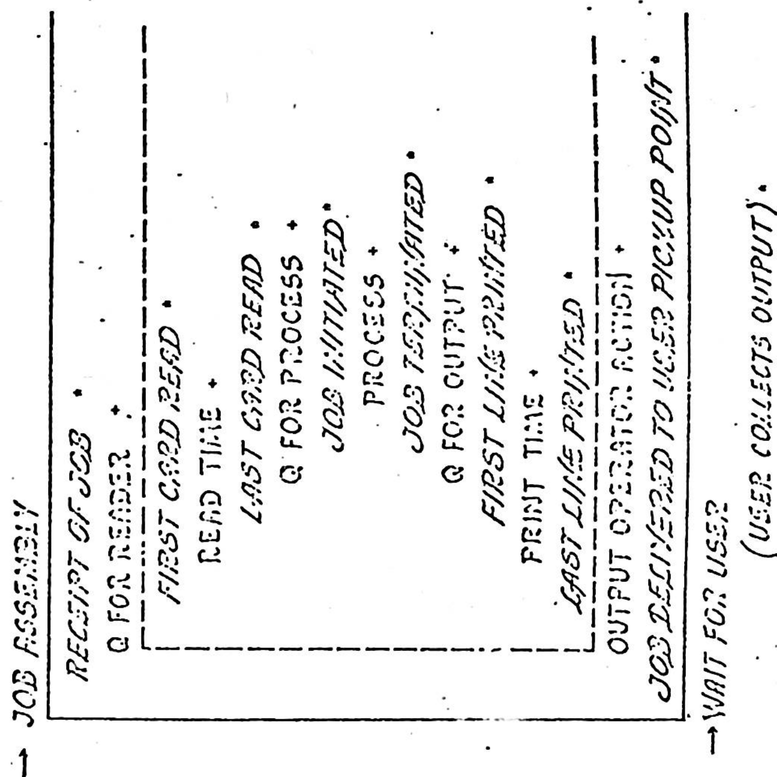
– – – – A = AV. OF LAST 4 WEEKS

TREND REPORTS
~ CLASS A TURNROUND ~ UNSMOOTHED



MONTHLY REPORT OF SERVICE TO CLASS A USERS
JANUARY, 1976





A HYBRID COMPUMETRIC TOOL

OR THE SYMBIOTIC RELATION OF A HARDWARE AND A SOFTWARE PERFORMANCE MONITOR

by Mr. Lawrence Pfeffer*, A. Rona* and A. Ressler+ - ~~August 1976~~

Abstract

In the environment of complex computer systems, software and hardware performance monitors are beginning to fill the compumetrician's tool bag. The former are programs, or implants, which observe selected aspects of system behavior and log their observation on suitable files for subsequent editing, analysis and report generation. Hardware monitors, on the other hand bear more similarity to conventional instruments in the sense that they are fabricated from electrical and mechanical components and are physically attached to the systems they measure in order to obtain performance data by observing the logical state of externally accessible protrusions on the spacimen system. Hardware monitors are similar to their software counterparts in the sense that they too write performance data onto a log file for subsequ data reduction.

Experience shows that both types of instruments suffer from limitations inhere in the techniques they use in order to acquire performance data. In particular hardware instrumentation has difficulty measuring such performance attributes as multiprogramming factor, paging rate and input/output rate. Software tools on the other hand, can't accurately measure such behavior indicators as the high-speed buffer hit ratio, and the percent of time the CPU is executing in Supervisor and Masked States.

This paper discusses a hybrid measurement technique, that is simultaneous use of a hardware and software performance monitor.

* ILTAM , + MALAM

1. Introduction

In large modern computing systems which support concurrent batch and interactive work-loads and create the illusion of virtual storage architecture, the Central Processing Unit is often busy a hundred percent of the time, the major portion of which is spent in the Supervisor State. In simplistic terms this means that the CPU is occupied most of the time in the performance of operating system overhead functions. In the framework of such systems, construction of software compumetric tools for even approximate determination of the amount of time the system is executing in Supervisor State is at best a difficult task. This sceptical statement is based on the observation that in the typical situation described above, the operating system is generally "modest" or "bashful" much of the time and circumvents exploration of its behavior by software tools which gather information via sampling techniques. The explanation for this is rooted in the fact that in practice sampling methodologies depend on a random background noise of interrupts, which have the beneficial side effect of storing system status in known and acessible memory locations at fundamentally random time intervals. Software tool designs often assume that they have at their disposal a pseudo-random interrupt source, which would then permit system behavior inferrence from an ensemble of random samples. It is often overlooked that if the interruption source is interfered with and thereby becomes non-random, then the sampling methodology becomes irreparably corrupted. This phenomenon has been described elsewhere [1] and has been termed the blind-spot phenomenon.

Realizing the existence of this problem an attempt was made to test the sensitivity of sampling techniques to the nature of the sampling source, and the experiment yielded significantly different results depending on the source used [1]. This paper describes a hybrid hardware/software experiment designed to shed light on the magnitude of errors which certain types of software based performance measurements may be subject to.

A hardware performance monitor is an instrument specifically designed to measure complex computer system behavior and may be attached to various extruding points[1] of the CPU or its peripheral organs. The government of the State of Israel and ILTAM recently acquired a basic hardware monitor[2], namely a TESDATA 1020-D Microsum monitor. The instrument has been designed to accurately measure "bread and butter" performance indicators, such as the percent of time the CPU is executing in Supervisor State, peripheral device busy percentages as well as the degree of overlap between the activity of various system components. In addition, the monitor can measure certain types of system behavior which are difficult or even impossible to measure with software tools[3][2].

The monitor senses the behavior of a specimen system via a series of "sense points" or pins which extrude from various components of computer systems. Although large computer systems may have hundreds of such pins, no more than one or two hundred are of practical interest to a compumetrician. Those pins which have useful operative meaning to someone interested in conducting a performance measurement study have their logical function and electrical characteristics documented. The documentation is such that it enables planning a hardware monitor based performance study by personnel without training in electronics.

---

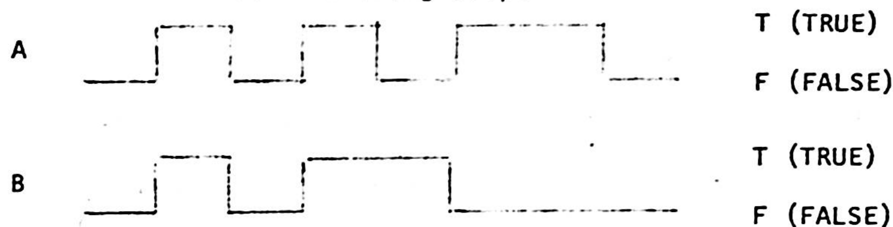
[1] the same points to which maintenance personnel attach oscilloscope leads during malfunction diagnosis

[2] the monitor's features have been considerably enhanced by ILTAM via a custom designed and built "front-end"

[3] examples are: the percent of time the CPU executes in masked state, the degree of CPU "lock-out" due to memory contention with the channels

In the framework of a given experiment, up to twenty sensor cables are connected to as many sense-points on the various system components under observation. The signals sensed are two-level, in the sense that they are either TRUE or FALSE and they may be combined by logic circuits (AND, OR, NOT, FLIP-FLOP and DECODE) on the monitor's patch-board. Accordingly the concurrence of two events (eg. CPU in Problem State and Channel 5 busy) may be derived by AND-ing their respective signals.

Signals may be analyzed by a hardware monitor in a number of distinct manners, however the Microsum monitor operates in what is called the time mode, which in effect measures the amount of time a signal is TRUE [2]. Thus if signals A and B have the following shape



then duration(A) / duration(B) = 4/3

Another useful signal analysis technique, count mode, counts the number of times a signal goes from FALSE to TRUE, which in the prior example would yield

count(A) / count(B) = 3/2

were the analysis technique available on the Microsum.

The signal levels TRUE and FALSE are different voltages, with specific voltage levels depending on the nature of components from which a given system is fabricated . In multi-component systems a variety of voltage levels are observed and they must be made to be compatible with each other and with the monitor's internal logic before they are sent there for analysis.

Up to ten sensor cables are fed into a concentrator box (two boxes are available) in order to equalize voltage levels of the various signals and

to "concentrate" the ten sensor cables into a single cable. The twenty sensed signals are fed into the monitor's back panel via the concentrator cables and are available at the twenty patch-board input hubs. The patch-board is a convenient place to access the input signals, to combine them via the various logic circuits (also having patch-board hubs), and to route the resulting signals to one of sixteen output hubs. The truth values (TRUE or FALSE) at the output hubs are periodically sensed by the monitor's hardware at a user specified rate not exceeding 16,384 time a second. Each sample takes a forty nanosecond "window" superimposed on the signal to be analyzed. If the signal is TRUE during the sampling interval (40 nsec) then the corresponding accumulator is incremented by one. After 32,768 sampling cycles the sixteen accumulators' content is displayed on the monitor's CRT and is simultaneously logged on a standard nine-track tape drive connected to the monitor. Subsequently the accumulators are reset to zero and the sampling procedure begins again.

The monitor's CRT provides immediate feedback on the status of an experiment, however the performance of the system measured is determined from suitable reports produced off-line from the tape log's content.

3. Software Monitors

IBM System/360-370 type Central Processing Units may be in one of four major states, as follows:

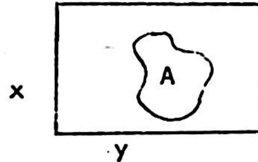
PROBLEM STATE	SUPERVISOR STATE
WAIT STATE	MANUAL STATE

where Manual State is the time spent with the CPU stopped (usually after the STOP key is depressed on the console), and tends to have a negligible value while a system is operational.

On 360-370 type processors there is a composite state indicator called the Program Status Word (PSW) [3], which identifies the CPU's major instantaneous states, including Wait (vs. Active), Problem (vs. System), the address of the next instruction to be executed and indications of CPU interruptability by its numerous peripheral organs. The concept of a PSW is augmented by the notion of a vector of Old PSW images stored in the computer's memory at pre-assigned memory locations corresponding to various interrupt classes (eg I/O, External, Machine Check).

A standard software technique for assessing the percentage of time the CPU is in either of its three major states is based on periodically sampling the content of one of the PSW images, which is generally the Old External PSW image. The algorithm is in effect a superimposition of two pseudo-random events, namely the random distribution of External interrupts (which trigger updating of the External PSW image) and the random times when the PSW image is sampled. The measurement technique is in many ways similar to a Monte Carlo method used to measure the area of irregular objects. The algorithm for measuring the

area of A is as follows:



A pin is randomly thrust into the square and a count is kept of the number of tries (N), and of the number of times the pin lands within the shape A (T). If the pin is guided by non-purposive guidance, then

$$\text{AREA OF A} = \lim_{N \to \infty} \frac{T}{N} * (x * y)$$

In practice, a reasonably accurate (>99 %) value for the area of A may be obtained by a few thousand "throws" of the pin.

Due to the increasing reliance of software performance monitors on the convenient technique of sampling (as opposed to event monitoring [1]) it was decided to put the practice of CPU state sampling to the test. The fundamental assumption governing the experiment was that if the sampling technique is indeed reliable, then simultaneous analysis of two different sources of PSW images should yield similar results. A series of experiments were conducted, using the External and I/O PSW images, with the following results:

$$S \approx S' + 14\,\%$$
$$P \approx P' - 7\,\%$$
$$(P+S) \approx (P' + S') + 7\,\%$$

where P and S designate Problem and Supervisor States, and the primed indicators designate values derived from the secondary PSW image source.

The sharp difference between the results of the two measurements showed that the assumed randomness in PSW image updating and/or the instance when the sampling algorithm obtains control is false. It was suspected that the inaccuracy is due to execution in the Masked State, and that the sampling algorithm was synchronized with critical aspects of system behavior in at

least two ways:

A - if the algorithm is scheduled to be activated at a time when the CPU is executing in Masked State, then the sampling cycle (and consequently a sample) will be delayed until the CPU exits from Masked State,

B - the two sampled sources (the External and I/O PSW images) appear to be stored by a random background noise of interrupts. This illusion is clearly false, since interrupts are often forced to wait until the CPU exits from Masked State.

It was assumed that the CPU cannot be in both the Problem and Masked States simultaneously[1], therefore it was anticipated that

A - if a CPU is in the Problem and Wait States P and W percent of the time, respectively, then P + W percent of the samples will be truly random: that is they will not occur during Masked State and will show the system to be in Problem and Wait States P and W percent of the time.

B - within this framework the system will be in Supervisor State S = 100 - (P + W) percent of the time, assuming zero percent Manual State. Masked State execution will clearly be $M \leq S$. As a result M percent of the samples will be interfered with and will be postponed until CPU entry into one of the following States:

WAIT, PROBLEM or SUPERVISOR AND NOT MASKED.

This in effect means that as a first order approximation the following distorted values will be reported by a sampling based monitor:

```
PROBLEM STATE = P + (M * P) / (100 - M)
SUPERVISOR ST = (S - M) + (M * (S - M)) / (100 - M)
WAIT    STATE = W + (M * W) / (100 - M)
```

---

[1] the act of interrupt mask removal is a privilaged operation, that is it requires that the system be in Supervisor State. Consequently, a Problem State system executing in Masked State could never be unmasked, nor could it enter Supervisor State.

It has been observed that the system on which the experiments were conducted is rarely in the Wait State during prime shift. The direct consequence of this is that the Problem State utilization should be further exaggerated by the measurement technique at the expense of Supervisor State, which should therefore appear to be lower than in reality.

4. <u>The Hybrid Test</u>

A hybrid test was designed to shed light on what was suspected to be a fundamental problem with certain applications of sampling techniques. Two experiments were set up, as follows:

1. <u>Hardware Monitor</u>:

   Sensors were attached to the following sense points of the IBM System/370 model 168:

   Masked State Execution

   CPU Busy

   CPU in Problem State

   CPU in Manual State

2. <u>Software Monitor</u>:

   A relatively simple software monitor was written [1] to collect the following performance data relevant to this experiment:

   CPU busy according to the two PSW image sources

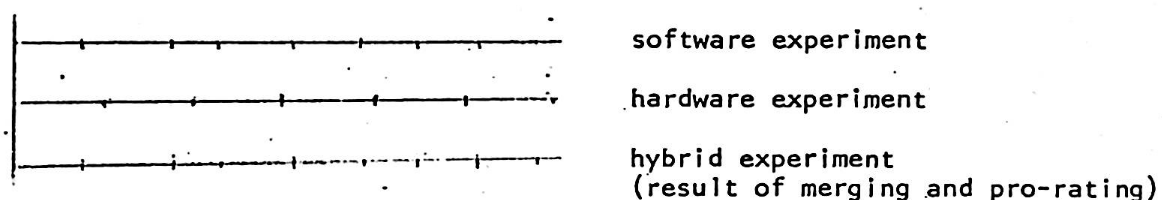
   CPU in Supervisor State according to the two PSW image sources

The hardware monitor was set up to log its observations once every 2.1 seconds (the minimum recording interval). The software monitor was directed to make a hundred observations, to log the time stamped results, to wait one second (in order to relinquish control to the rest of the system) and to repeat the measurement-recording-wait cycle a large number of times.

The software tool was used to gather I/O transfer counts per unit time interval in addition to CPU utilization values. This required operation of a high overhead detailed Supervisor Call trace program: GTF during the experiment. The combined effect of the software monitor and GTF placed an unusually high load on the system, resulted in higher than normal Supervisor State mode utilization, and in parallel yielded an unusually low Masked State execution.

5. Synchronization Problems

Even if it were possible to synchronize the two experiments at their outset, the series of successive observations would inevitably get out of synchronization due to the difficulty of guaranteeing that the software monitor should log its information at regular intervals.[1] A practical approach to overcoming this problem is to merge the results of the two experiments (assumed to be synchronized at start-up time), pro-rating the hardware monitor data where necessary. Thus, the following timing relationships may be observed:



Unfortunately perfect initial synchronization is not straightforward. The two experiments operate within their own time frame, with respect to experiment start time, and this necessitates another level of synchronization at the start of the experiment. Various methods have been tried, but the technique finally adopted consisted of starting both experiments at random times relative to each other, then stopping the CPU for approximately ten second with the experiments already in progress. During the "idle" time the hardware monitor would of course maintain sampling the CPU state and would record a number of identical observations, corresponding to one of the following possibilities:

100 % Wait State and 100 % Supervisor State

0 % Wait State and 100 % Supervisor State

0 % Wait State and 100 % Problem State

---

[1] The difficulty is due to the fact that the software experiment was designed to run on a large multi-access system, whose behaviour could not be altered (by definition of a behaviour observing experiment)

In either case a detailed plot would show a clearly recognizable synchronization signal lasting a number of sampling intervals. In parallel, while the CPU is stopped, the software monitor (as well as all processes in the CPU) is placed in a state of suspended animation, and as a result two successive records will be time stamped with an approximately ten second time difference. Inspection of a detailed time stamped plot permits detection of the start of the two experiments, and enables start-synchronization of the two measurements on the hybrid-log. Samples of the detailed plot may be found in Appendix I and II of this paper.

6. Result of the Hybrid Experiment

The technique chosen to exhibit the results of the hybrid experiment was to plot a detailed time-sequence graph of the various behaviour indicators. In order to aid in visual detection of underlying relationships, the variables were smoothed via a five element moving average technique before they were plotted. Appendix I and II contain portions of the detailed graphs, before and after smoothing.

The plots indicate that there is indeed a significant error in the value of both the Problem and Supervisor State utilizations, as derived by a simple Monte Carlo software sampling technique. Although the plots of the hardware and software behaviour data are slightly out of synchronization (the software "clock" lagging behind by approximately seventeen seconds) it is clear that there is a definite relationship between the two types of measurements.

The following is the list of utilization values reported by the hybrid experiment:

	Monitor	Ext.PSW	I/O PSW
System State	87.6	48.7	56.3
Problem State	12.1	51.1	43.6
Masked	24.9	-	-
Wait State	.3	.2	.1

It has not been possible to derive a simple relationship between the hardware and software measurements, however the compounded effects of the blind-spot phenomenon, the Uncertainty Factor introduced by the altered system behaviour due to the experiment, and the non-random displacement of sampling (observation)cycles appear to be much more complex than the hypothesis formulated in section three of this paper. The experiment clearly illustrated the feasibility of conducting hybrid performance studies, showed that sampling based software monitoring techniques have an inherent weakness and that the hypothesized blind-spot phenomenon is indeed a reality.

References

[1] Pfeffer L. and Ressler A., "Practical Compumetric Tools For Use in Virtual Storage Systems", ILTAM Corp. for Planning and Research, Ltd., Tecnical Report Two - 1975 and Proceedings of IPA (ILA) Conference 1975 pp. 429 - 476.

[2] Pfeffer L. and Loev D., "Hardware Monitor Technical Note 1", ILTAM Corporation for Planning and Research, Ltd., Oct. 1975.

[3] IBM System/370 Principles of Operation, form GA22-7000 IBM Corporation. 1972.

APPENDIX I : SAMPLE HYBRID MONITOR OUTPUT -- UN-SMOOTHED DATA

UN-SMOOTHED DATA

76225	4154054
76225	4154287
76225	4154485
76225	4154664
76225	4154872
76225	4155077
76225	4155274
76225	4155465
76225	4155650
76225	4155848
76225	4156050
76225	4156243
76225	4156479
76225	4156682
76225	4156877
76225	4157068
76225	4157259
76225	4157463
76225	4157651
76225	4157835
76225	4158558
76225	4159155
76225	4159360
76225	4159570
76225	4159773
76225	4159958
76225	4160207
76225	4160437
76225	4160620
76225	4160822
76225	4161010
76225	4161191
76225	4161384
76225	4161576
76225	4161764
76225	4161957
76225	4162136
76225	4162311
76225	4162503
76225	4162688
76225	4162879
76225	4163069
76225	4163264
76225	4163493
76225	4163700
76225	4163890
76225	4164094
76225	4164292
76225	4164489
76225	4164678
76225	4164880
76225	4165076
76225	4165265
76225	4165458
76225	4165656
76225	4165861

Software Synch

Monitor Synch.

NOTE THAT HARDWARE MONITOR DATA IS SLIGHTLY FORWARD SHIFTED IN TIME

Symbols used in the hybrid monitor plots:

'	CPU^{M}_{MASK}
+	A REPRESENTATIVE CHANNEL BUSY
S	CPU^{X}_{S}
O	CPU^{I}_{S}
P	CPU^{X}_{P}
A	CPU^{I}_{P}
*	CPU^{M}_{P}
-	CPU^{M}_{S}
#	$\lvert CPU^{X}_{S} - CPU^{I}_{S} \rvert$
>	$\lvert CPU^{X}_{P} - CPU^{I}_{P} \rvert$

where CPU^{X} ...... CPU state according to external old PSW

CPU^{I} ...... CPU state according to I/O old PSW

CPU^{M} ...... CPU state according to real PSW (monitor)

CPU_{S} ...... CPU in Supervisor State

CPU_{P} ...... CPU in Problem State

SMOOTHED DATA

76225	41[illegible]
76225	4154485
76225	4154664
76225	4154872
76225	4155077
76225	4155274
76225	4155465
76225	4155650
76225	4155848
76225	4156050
76225	4156243
76225	4156479
76225	4156682
76225	4156877
76225	4157068
76225	4157259
76225	4157463
76225	4157651
76225	415[illegible]35
76225	415[illegible]53
76225	4159155
76225	4159360
76225	4159570
76225	4159773
76225	4159958
76225	4160207
76225	41[illegible]437
76225	41[illegible]20
76225	41[illegible]22
76225	41[illegible]
76225	4161191
76225	4161384
76225	4161576
76225	4161764
76225	4161957
76225	4162136
76225	4162311
76225	4162503
76225	4162698
76225	4162879
76225	4163069
76225	4163264
76225	4163493
76225	4163700
76225	4163890
76225	4164094
76225	4164292
76225	4164489
76225	4164678
76225	4164880
76225	4165076
76225	4165265
76225	4165458
76225	4165656
76225	4165861
[illegible]	[illegible]6046
[illegible]	[illegible]6242
[illegible]	[illegible]6427

← SOFTWARE SYNCH.

← MONITOR SYNCH

NOTE THAT MONITOR DATA IS SLIGHTLY FORWARD SHIFTED IN [illegible]

שיטת העלות הכוללת להערכת הצעות

למכרז של משרד התקשורת.

כללי.

ביולי 1973 פירסם משרד התקשורת מכרז להרחבת הכח החישובי של מערכת המחשבים שהייתה מותקנת במרכז לשירותי מידע בתל-אביב.

ועדת מכרזים בראשות מנכ"ל המשרד דאז מר שמחה סורוקר הטילה על צוות עבודה להתוות מראש את תהליך קבלת ההחלטה בדבר בחירת ספק הציוד. תהליך זה, אשר כלל את כל השלבים והקריטריונים לפיהם יפעלו חברי ועדת המכרזים, נקבע מראש לפני פתיחת המעטפות. נראה לועדה, ובצדק, שבועדות מכרזים ממשלתיות שיטה זו חיונית להבטחת מידה מקסימלית של אוביקטיביות בבחינת ההצעות.

ועדות קודמות.

עיון במסמכים על עבודתן של ועדות מקבילות קודמות הראה כי הכלים המקובלים היו שיטות ניקוד למיניהן, הווה אומר, יחוס משקלות לתכונות של הציוד, התוכנה, וכו'. בשלב הערכת ההצעות אנשי הועדות נתנו ציונים לכל אחת מהתכונות ולאחר מכן נעשה שקלול של התכונות.

לדוגמא:

אחת הועדות חילקה את ההחלטה למספר פרקים עיקריים כגון: תוכנה, ציוד שרות וכו'. כל אחד מפרקים אלה קיבל משקל מסוים באחוזים מכלל ההחלטה. נביא דוגמא היפותטית.

תוכנה	30%
ציוד	30%
תפוקה	20%
שרות	20%
סה"כ	100%

פרקים מתחלקים חלוקה נוספת, לדוגמא, נחלק את פרק התוכנה לסעיפים הבאים:

תוכנה לעיבוד מכלולים	40%
תוכנה לתקשורת נתונים	20%
תוכנה למסדי נתונים	20%
תוכנה להסבת מערכים	20%
	100%

גם סעיפים אלה מתחלקים חלוקה נוספת, לדוגמא, נחלק את התוכנה לעיבוד מכלולים כדלקמן:

מהדר COBOL	20%
מהדרים אחרים	20%
תוכנית מיון	20%
תוכניות עזר אחרות	20%
כרטיסי בקרה	20%
סה"כ	100%

אפנה את תשומת לבכם שבדוגמא שבניתי כאן גורם כמו המהדר ל- COBOL מהווה 2.4% מההחלטה. עובדה זו בראית באופן אינטואטיבי מתקבלת על הדעת בשיטה זו של-הערכת ההצעות.
שיטות כגון זו בפסלו על ידי המרכז לשרותי מידע. השיקול העיקרי לפסילתן היה החשש שעלול להווצר מצב לאחר פתיחת המעטפות של ההצעות שגורמים שונים אשר קיבלו משקל קטן יחסית, יהיו בעלי השלכות חמורות באשר לצורת הפעילות של המרכז לשרותי מידע. למרות השלכות אלו לא תהיה אפשרות ליחס לגורמים הנ"ל את המשקל הנכון בלי לסטות ממערכת הכללים שהוסכם עליה מראש.

כדי להמחיש את האמור בשתמש בדוגמא הקודמת; נניח שבאחת ההצעות מוצעת מערכת עם מהדר ל- COBOL שאיננו תומך במספר אופציות כמו למשל REPORT WRITER. במרכז לשרותי מידע נעשה שימוש ברחב ויעיל ביותר בכלי זה ומוערך שהעדרו יגרום להוספת 4 תוכניתנים לצוות התכניתנים הקיים. בתהליך מתן הניקוד להצעה ניתן "להעניש" את ההצעה באחוז אחד או שניים לכל היותר מפני שמהדר ה- COBOL כולו מהווה 2.4% מההחלטה וכמה כבר ישקול גורם כמו ה- REPORT WRITER שהוא אחת מן האופציות הרבות של שפת ה-COBOL ?.

4 תוכניתנים עולים כ-25 אלף לירות לחודש וברור שאין ליחס להם רק 1%-2% ממשקל ההחלטה...

בעיות נוספות שנתגלו בשיטות הניקוד השונות נבעו מהאופן בו הוכנסו גורם המחיר ותפוקה למכלול השיקולים וכמו כן הוברר שבעבר נהגו חברי ועדות שונות לבחור את ההצעה הנראית להם לפי שיקולים אחרים. לאחר מכן, בשלב מתן הניקוד על ידי כלל הועדה דאגו "לעגל" את המספרים לכוון ההצעה עליה החליטו מראש. כאשר השיקולים למתן הציונים הם בלתי מוגדרים הרי ברור שכמעט כל ציון יכול להתקבל על הדעת. אינני מופתע שחברי ועדות נהגו לעשות כן כי הרי שיטות הניקוד למיניהן מרחיקות את ההחלטה מהמציאות בשטח ומעבירות אותה לפסים אינטואיטיביים, במצב זה מעדיף איש הוועדה להחליט קודם ולאחר מכן ל"הלביש" את ההחלטה בלבוש "האוביקטיבי" של שיטת הניקוד.

<u>השיטה שנקבעה.</u>

עובדה שיטה להערכת ההצעות הבנויה על שלשה שלבים:

- פסילה על הסף.
- ריצות ניסוי.
- חישוב העלות הכוללת של מש"מ.

להלן אציג בצורה תמציתית את העיקרים של כל אחד מהשלבים:

3.1 <u>שלב א'</u>

פסילה על הסף של הצעות שלא יעמדו באחד מתנאי המינימום הבאים:

- מילוי אחר הדרישות הפורמליות של המכרז כנדרש בחוק.
- תפוקה התחלתית מינימלית של המערכת המוצעת של פי 1.70 מהמערכת הקיימת.
- תוכנה מינימלית כפי שהוגדרה במסמכי המכרז.
- אפשרות לביצוע הסבה חלקה בלי פגיעה ברציפות השרותים הניתנים על ידי מרכז החישובים.
- ידע ונסיון של הספק בסוג המערכת המוצעת.
- אפשרות לתת גיבוי לעיבודים רגישים.
- אפשרות להרחיב את הציוד כך שיאפשר את המשך הפעילות של מש"מ במשך 5 שנים ללא הסבות. הגידול הצפוי בצריכת זמן מחשב במשך 5 שנים הוערך ב-300%.

3.2 שלב ב'

בשלב זה יבוצעו ריצות ניסוי. מטרות ריצות הניסוי יהיו קביעת התפוקה הצפויה מהמערכות המוצעות. המערכות המוצעות ינוסו בקונפיגורציות שונות. באופן זה יקבע מה יהיו השלבים של הרחבת הציוד עם גידול צריכת זמן המחשב ובאילו נקודות על ציר הזמן יבוצעו.

תוצאות הריצות יהיו רשימת קונפיגורציות מומלצות וגרף המתאר את ההוצאות ברכישת/שכירת ציוד לאורך 5 שנים מתאריך ההתקנה.

3.3 שלב ג'

בשלב זה תחושב העלות הכוללת של מש"מ.
העלות הכוללת של מערכת מוצעת בתונה היא הסכום של שני מרכיבים:

א. ההוצאות הישירות עבור ההתקנה, ההסבה והאחזקה של המערכת המוצעת לאורך 5 שנים.

ב. מרכיבי העלות בתוך מש"מ שיכולים להיות מושפעים מהמערכת המוצעת.

3.3.1 ההוצאות הישירות.

ההוצאות הישירות יחושבו לפי שכירות וקניה והסכומים יעברו הוון לפי שער ריבית גבוה מהמקובל. הוחלט שהשימוש בשער ריבית גבוה בהוון ההוצאות משקף את הרצון של הועדה לתת משקל נמוך יחסית בהחלטה להוצאות שיבוצעו בשלבים מאוחרים של התכנון כי הרי שלבים אלה עלולים לעבור שינויים רבים.

סעיפי העלות שילקחו בחשבון הם:

- מחיר הציוד.
- מחיר התוכבה.
- מחיר האחזקה של הציוד והתוכנה.
- מחיר ההסבה.
- מחיר השינויים בתשתית (חדר מחשבים, מיזוג אויר, חשמל וכו').
- מחיר ההובלה, מסים ובטוח.
- הוצאות הדרכה (תשלומים עבור הקורסים דזמן האדם שיושקע).
- שונות.

3.3.2 מרכיבי העלות בתוך מש"ם.

ילקחו בחשבון ההשפעה של ההצעה על כמות ואיכות כח האדם שידרשו לפונקציות מסויימות (פיתוח, הפעלה, יצור) וטיב השרות של מש"מ ללקוחותיו. אלה נקראו גורמים מושפעים. לגבי יתר המרכיבים (חומרים, ניקוב, שרדתים, תפעולים, אמרכלות, תקורה וכו') סוכם מראש כי ההשפעה תהיה שולית וזניחה.

מראש נקבע שלצורך הערכת העלות של הגורמים המושפעים נביא בחשבון רק את הגורמים המשפיעים הבאים:

- תכונות הציוד.
- תכונות ומגוון התוכנה.
- תפוקה אפקטיבית בעיבוד מכלולים.
- יכולת לפתח ולישם אפליקציות בתקשורת.
- צורת ההפעלה של המערכת.
- אמינות הציוד והתוכנה.
- צורת הגיבוי.
- מודולריות ההרחבה של המערכת במקרה של חריגות מהתיכנון.
- תמיכת הספק.

עבור כל הצעה תוכן טבלה בה יוצגו הגורמים המשפיעים מול הגורמים המושפעים ולכל משבצת תעשה הערכה כספית.

הטבלה תראה באופן כללי כך:

גורמים מושפעים / גורמים משפיעים	עלות כח אדם בפיתוח	בהפעלה	ביצור	פרס עבור רמת שרות
תכונות הציוד				
תכונות ומגוון התוכנה.				
תפוקה בעיבוד מכלולים				
תמיכת הספק.				

מילוי טבלה זאת איננו מלאכה קלה. כדי להבהיר באיזה סוג של שאלות נאלצת הועדה לטפל בתהליך קבלת ההחלטה הזאת אציג כאן כמה דוגמאות:

<u>שאלה א'</u>: האם יש-הבדל בין הצעה א' להצעה ב' במידת ההשפעה שלהן על כח האדם הנחוץ להפעלת המערכות? אם כן, מהו ההבדל ובמה הוא מתבטא מבחינת העלות של כח אדם במשך 5 שנים.

<u>שאלה ב'</u>: נניח שמערכת מוצעת בעלת תפוקה גבוהה מעבר למה שנראה נחוץ לפי תכנית הגידול של משיים. מה ההשפעה של עובדה זו על כח האדם לפיתוח מערכות? הועדה תאלץ להתמודד עם שאלה זו ולקבוע הערכות כספיות לחסכון הצפוי בכח אדם.

<u>שאלה ג'</u>: מה ההבדלים בין שתי הצעות מבחינת השפעתן על טיב השרות שמקבלים לקוחות משיים ואם קיימים הבדלים משמעותיים, כמה כדאי לשלם עבור התכונות של מערכת מסויימת על פני השניה.
לאחר מילוי ערכי הטבלה הם יסוכמו עם מרכיבי ההוצאות הישירות ובאופן זה תחשב העלות הכוללת. ההצעה בעלת העלות הכוללת הנמוכה ביותר תזכה במכרז.
לאמיתו של דבר אין צורך לחשב את העלות הכוללת עד הסוף; כדי להשוות בין שתי הצעות מספיק לבדוק עבור כל משבצת בטבלה אם יש הבדלים בין שתי ההצעות. לאחר מכן מעריכים הערכה כספית רק את ההבדלים בין שתי ההצעות, ומשוים את סכום ערכי הטבלה המתקבלים עם ההפרש בין ההוצאות הישירות של שתי ההצעות.

<u>סיכום</u>.

לשיטה המוצגת כאן יש חסרון אחד בולט; היא מחייבת התעמקות והבנה מפורטת של ההצעות המוצעות. לא קל לענות על שאלות כדוגמת אלה מבלי להכיר היטב את הארגון בו תופעל המערכת ואת כל הפרטים במערכת המוצעת שיכולים להיות בעלי השפעה על תהליכי העבודה.

מצד שני בולט גם היתרון של השיטה:
ההערכות נעשות לפי שיקולים כלכליים של עלות ותועלת ולא לפי שיקולים אינטואיטיביים בלתי מוגדרים.

קבוצת דיון מס' 4:

"יישומי מחשבים ברפואה"

יו"ר: ד"ר ש. פנחס

ז. קריים

ש. פנחס

ש. ביילים

עיבוד תוצאות מעבדה כעזר לאבחנה וטיפול רפואי

I. מבחני תיפקוד יתרת הכליה

ז. קריים

מכון אנדוקריני, מרכז רפואי ע"ש חיים שיבא, תל-השומר

א. רייכרט

אוניברסיטת בר-אילן

ד. מוצקין

אוניברסיטת חיפה

מיגוון הבדיקות העומד היום לרשות הרופא הוא כה נרחב ולפעמים מורכב שאף המומחה מתקשה לעיתים בקביעת מירב המסקנות המתבקשות מהנתונים המעבדתיים (1). מתוך מגמה להתגבר על בעיה זו, השתמשנו במערכת לאינטרפרטציה של תוצאות בדיקות מעבדה (2) אשר פותחה כחלק מפרויקט כללי של תוכנית אבחנתית בעזרת מחשב (3,4,5), פרי שילוב בין המרכז המדעי י.ב.מ. בחיפה והמכון האנדוקריני בתל-השומר. מערכת זו (2) מעבדת תוצאות מעבדה כעזר לרופא הן בשלב האבחנתי והן במעקב אחרי תגובה החולה לטיפול הניתן לו. עיין ב-(2) לשם תאור המערכת הכללית.

ברצוננו להציג כעת יישום המערכת בשטח רפואי ספציפי: מבחני תיפקוד יתרת הכליה. בדיקות אלה כוללות:

1. Urinary 17OHCS -- basal
2. Urinary 17KS --basal
3. Plasma cortisol-- basal / diurnal variation
4. Urinary cortisol metabolites (THE+THF+alloTHF+F) --basal
5. Urinary corticosterone metabolites (THB+alloTHB+THA) --basal
6. Urinary compound S metabolite (THS) --basal
7. Urinary DOC metabolite (THDOC) --basal
8. Urinary pregnanediol--basal
9. Urinary pregnanetriol--basal
10. Urinary pregnanetriolone-- basal

11. Single dose dexamethasone test--plasma cortisol response
12. Small dose dexamethasone test--plasma cortisol response
13. Small dose dexamethasone test--urinary 17KS response
14. Small dose dexamethasone test--urinary 17OHCS response
15. Large dose dexamethasone test--urinary 17OHCS response
16. ACTH stimulation test--urinary 17OHCS response
17. ACTH stimulation test--plasma cortisol response
18. ACTH sustained stimulation test--plasma cortisol response
19. ACTH sustained stimulation test--urinary 17OHCS response
20. Metyrapone test (pituitary ACTH reserve test)

המבחנים הנ"ל בודקים את תקינות הציר היפותלמוס-היפופיזה-יותרת הכליה. הם משמשים כעזר לרופא בשלבי אבחנה וטיפול של מצבי יתר-פעילות או תת-פעילות של הציר. בדיקות 1,2 משקפות את רמת כלל ההורמונים המופרשים מקליפת יותרת הכליה. מבחן מס. 3 קובע את רמת הקורטיזול, ההורמון הפעיל ביותר מבחינה ביולוגית מתוך כלל ההורמונים שמקורן בקליפת יותרת הכליה. בדיקה זו גם קובעת את שינויי רמות הקורטיזול במשך היממה. מבחנים 4-10 בודקים את הרמה הבסיסית של הורמונים שונים אשר נוצרים בתהליך הביוסינטזה של סטרואידים ביותרת הכליה. בדיקות 11-15 מהוות מבחנים דינמיים אשר מטרתם לבחון את המשוב (feedback) שבין הבלוטות האנדוקריניות היפותלמוס-היפופיזה ויותרת הכליה. בדיקות 16-19, גם הן מבחנים דינמיים, משקפים את כושר יותרת הכליה להגיב לגירויי הורמונלי. המבחן האחרון, מס. 20, בודק את עתודת ה- ACTH , ההורמון אשר משמש כגירוי פיזיולוגי של בלוטת יותרת הכליה. יתר על כן, כל פגם בציר היפותלמוס-היפופיזה-יותרת הכליה יתגלה במבחן זה.

נדון בהרצאה בקשיים שנתקלנו בפיתוח הנוסחאות המתאימות להגדרת כל מבחן ומבחן, ואשר קיבלו את צורתן הסופית לאחר תהליך מייגע של תיקון ושיכלול מתמיד בעזרת תוכניות האישור ועידכון שבתוך המערכת. הגדרות אלו עמדו במבחן בסידרות מקיפות של בירורי תקיפות (validation).

נביא כעת שתי דוגמאות להדגים שימוש המערכת בעיבוד תוצאות של המבחנים הנ"ל.

Lab Data

Enter date: Day Month Year 9 10 75*
Enter test ID: 13
Test number 13 is Small dose dexamethasone test --urinary 17KS response ?
(Y=Yes, N=No) Y
Enter patient ID: 6447218
Enter test data

Age	25
Sex	1**
Basal level	21
Response level	2

Normal 17KS suppression to small dose dexamethasone test.
High 17KS basal level.
17KS basal level borderline between normal and high.

Lab Data

Enter date: Day Month Year 10 11 75
Enter test ID: 14
Test number 14 is Small dose dexamethasone test -- urinary 17OHCS response
(Y=Yes, N=No) Y
Enter patient ID: 7823418
Enter test data

Age	30
Basal level	2
Response level	1

Inconclusive dynamic test
Low 17OHCS basal level

---

* תשובות המפעיל מודגשות בקו.
** קוד למין זכר.

הדוגמא הראשונה מראה שמתוך הנתונים המעבדתיים הופק מידע הן לגבי המבחן הדינמי, כלומר התגובה לגירוי פרמקולוגי, והן אודות הרמה ההורמונלית הבסיסית. בנוסף, התוכנית אינה מסתפקת בקביעת רמה הורמונלית כנורמלית, נמוכה או גבוהה אלא מסוגלת לסווג את הרמה הבסיסית בגבול שבין שלושת המיונים. באופן כללי, מספקת המערכת אינטרפרטציה מירבית מכל מבחן. דבר זה התבטא גם בדוגמא השניה המבליטה מצב שבו הנתונים אינם מאפשרים הכרעה באם התגובה למבחן היא משמעותית או לא.

הנסיון עד כה במבחני יתרת הכליה כדגם מלמד שהמערכת עונה לקריטריונים שהוצבו כמטרה על-ידי מפתחי המערכת (ראה 2.), דהיינו: היא גם קלה להפעלה ן, ביחס לרופא, עדיפה מבחינת מהירות, דיוק, עקביות והיקף האינטרפרטציות. יתר על כן, התברר במשך העבודה שבתוכנית טמון מכשיר רב ערך למטרות חינוך רפואי. נסיים במילת אזהרה: המבחן האמיתי של המערכת הוא בשימוש יום יומי בבית חולים, נסיון הנערך כעת. רק בסוף תקופת נסיון בשדה המציאות נדע באם המערכת הוכיחה את עצמה.

<u>הכרה תודה</u>

אנו מודים ל:-

פרופ' ב. לוננפלד (מכון אנדוקריני, מרכז רפואי ע"ש שיבא, תל-השומר ואוניברסיטת בר-אילן) עבור הדחף לפיתוח המערכת.

ד"ר ו. שילד וד"ר ב. גביש (מרכז מדעי י.ב.מ., חיפה) עבור תרומתם הרבה בפרויקט זה.

מרכז המחשבים באוניברסיטת בר-אילן העמיד לרשותנו את האפשרות הטכני לביצוע הפרויקט ועל כך נתונה תודתינו.

עבודה זו נתמכה ע"י מענקי מחקר מהמדען הראשי של משרד הבריאות לפרופ' ב. לוננפלד, מקרן פורד (מס. 470-67) ומהארגון הבריאות העולמי.

סיפרות

1. Raymond, S. Physician-oriented data processing. JAMA, 234, 83-85, 1975.
2. Motzkin, D., Kraiem, Z., Gavish, B. A study in computer-assisted interpretation of laboratory tests. Israel IBM Report. January 1976.
3. Schild, W., Gavish, B., Lunenfeld, B. Computer assisted diagnosis with an application to endocrinology. Israel IBM Report. April 1974.
4. Gavish, B., Schild, W., Lunenfeld, B. A diagnostically oriented APL shared variable system for outpatient care. Proceedings of the IFIP working conference for patient care. Amsterdam, 1975.
5. Schild, W., Lunenfeld, B., Gavish, B. A system for computer-aided diagnosis. Israel IBM report. February 1976.

השפעת הדיגיטליס על מרווח פעימות הלב בפרפור הפרוזדורים - אנליזה מחשבית.

ד"ר ש. פנחס   פרופ' ב. זיצ'ק

מחלקה פנימית א', ביה"ח האוניברסיטאי הדסה ויחידת המחשב ביה"ס לרפואה

של האוניברסיטה העברית, ירושלים.

נתמך ע"י הקרן לקידום האנושות

פעילות התקיבה של הלב תלויה במערכת גרויים חשמליים שמקורם בפרוזדור הימני. גל חשמלי שנוצר בדופנו של פרוזדור זה מתפשט לאורך קירות הלב במערכת הובלה מיוחדת וגורם להתכווצות שריר הלב. התכווצות תקינה זו מבטיחה שאיבת דם יעילה על ידי הלב. בדרך כלל נוצר הגל החשמלי שבעים פעמים בדקה אך במצבי מחלה שונים משתבשת הופעתו או הובלתו. דבר זה גורם קצב לב בלתי סדיר והתכווצות בלתי יעילה של שריר הלב. לקוי המכונה "פרפור פרוזדורים". בלקוי זה, הפרוזדורים כמעט שאינם מתכווצים וחדרי הלב מתכווצים בצורה בלתי סדירה דבר שבא לבטויו בדופק בלתי סדיר. בתנאים אלה לוקה שאיבת הלב, והדם אינו מתנקז כראוי. מצב זה מכונה אי ספיקת הלב. בטפול בלקוי זה נעזר הרופא בתכשיר הדיגיטליס אשר מאיט את ההולכה החשמלית בלב ובכך גם את קצבו ומייעל את התכווצות הלב ולכן שאיבתו משתפרת. יעילותה של תרופה זו טומנת בחובה סכנה רבה כי האטה יתר של קצב הלב יכולה להביא להופעת גרויים חשמליים בלתי רצויים בלב אשר עלולים להקטין את פעילותו.

בדי למנוע מבוך זה הידוע כ "הרעלת דיגיטליס" יש להקפיד על מינון מדויק של תרופה זו דבר שאינו פשוט כל עקר כי המינון אינו אחיד ומשתנה מחולה לחולה. על מנת לקבוע את המינון המיטבי לחולה נעזר הרופא

.../2

בשיטה כימית באמצעותה ניתן לקבוע את רמת הדיגטליס בדם. למרות זאת, ההתאמה בין רמה זו לבין השפעת התרופה על קצב הלב אינה חד ערכית, וזאת בעקר מפני שהבדיקה הכימית מבוצעת במרווחי זמן ארוכים יחסית (פעם עד פעמים בשבוע). מסבה זו סומך הרופא על קצב הדופק כמדד למידת השפעת התרופה תוך הנחה שכאשר קצב הלב יורד, סכנת ההרעלה מהתרופה עולה, ועל בן נוטה הוא להתאים את מינונה לפי קצב הלב. אך גם שיטה זו אינה יעילה, והחולה עלול להידרדר למצב של הרעלה.

מסבה זו פתחנו בחפוש אחר מדדים אחרים אשר יכולים להצביע בעקיפין על רמה בלתי רצויה של הדיגיטליס ובכך למנוע את סכנת ההרעלה. מדדים כאלה ניתן למצוא בתרשים של לב החולה (א.ק.ג.). אחד המדדים אשר הופק מתרשים החולה הוא אות מרווח הפעימה (Cardiac interbeat interval signal). מדגם של מרווחים בחולים השונים הוזנו למחשב לשם חשובים סטטיסטיים ומתמטיים שונים כמו התפלגות הספק (Power Spectrum), על מנת לגלות את המדד המותאם ביותר לרמת הדיגוקסין בדם.

חומר ושיטות

__החולים:__ הסקר מבוסס על 58 חולים עם פרפור פרוזדורים מסבה שגרונית או טרשתית, אשר אושפזו בביה"ח "הדסה", במחלקות שונות בתקופה 13.2.76 = 7.7.76. הסקר כלל 27 נשים ו-31 גברים, בגילים בין 75 - 24 שנים.

3/...

תרשימי ה-א.ק.ג. : הרשום ואלקטרוקרדיוגרפי נעשה על ניר רגיש לחום במכשיר (HP דגם 1511B) כשהענות התוד היא כ- 70 הרץ). בכל רשום נמדדו לפחות 33 גלי QRS מלאים וברורים (טבלה מס' 1).

רמת הדיגיטלים בדם: נקבעה ע"י היחידה לפרמקולוגיה קלינית בשיטת Radioimmunoassay. לפי שיטה זו ערכי הדיגיטלים התקינים נעים בין 2.4 - 0.8 ננוגרם ל- מיליליטר. לצורך החשובים נלקחו בחשבון רק החולים אשר בדיקת הדיגיטלים בדם נעשתה בסמיכות זמן לתרשים ה- א.ק.ג. 12 חולים אשר לא עמדו בתנאי זה לא נלקחו בחשבון הסטטיסטי הסופי.

אות מרוח הפעימה: מתקבל ע"י המרוח בין גלי R עוקבים (טבלה מס' 1, 2).

ספרור (Digitization): הזנת התרשימים לתוך המחשב בוצעה באמצעות מספרר קולי Graf/Pen Digitizer הקשור ל -15 PDP שליחידת המחשב של ביה"ס לרפואה.

שיטות חשוב: דגימות מרוחי הפעימה עובדו בשיטות סטטיסטיות רגילות בנוסף לכך הופעל עליהם אלגוריתם טרנספורם המהיר של פורייה Fast Fourier Transform (לפי שיטת Sand-Tukey) אשר מבצעת את הדצימציה (Decimation) בתחום התדירויות. תכנית זו יעילה ומהירה מאד. היא דורשת 875 מלים בזכרון ומבצעת במחשב 1800 - IBM הפיכה של 1024 נקודות ב- 0.8 שניה (טבלה מס' 2). במסגרת החשובים נבדקו תכונות שונות של התפלגות ההספק (Power Spectrum) לצורך חשובי המתאם השתמשנו

4/...

במרכז הכובד על ציר ה- Y של התפלגות ההספק.

תוצאות ודיון

כדי להדגים את דרך העבוד מובא תרשים א.ק.ג. טיפוסי (טבלה מס' 1) התפלגות אורך מרווח הפעימה (טבלה מס' 3) אות הפעימה והתפלגות ההספק (טבלה מס' 2) הפרמטרים החשובים שהופקו במחקר זה מובאים בטבלה מס' 4 בה מיוצגים 46 חולים.

רמת הדיגיטליס הממוצעת היתה 1.6 נ"ג למ"ל. ב-9 חולים נצפתה רמת דיגיטליס נמוכה מ 0.9 נ"ג למ"ל, וב-7 חולים נמצא ערך הגבוה מ 2 נ"ג למ"ל. 65% מהחולים הראו רמת דיגיטליס תקינה. קצב הלב הממוצע היה גבוה מהרצוי כ 100 פעימות בדקה במקום ב70 פעימות בדקה. למרבית ההפתעה נמצא שמקדם ההתאמה בין רמת הדיגיטליס והדופק מחד ורמת הדיגיטליס ומרווח הפעימות הממוצע מאידך, היה נמוך ובלתי משמעותי מכאן שהמדדים אלה אינם מצביעים על רמת הדיגטליס (טבלה מס' 5). מאידך ניתן ללמוד מהטבלה גם על התאמה משמעותית בין התנודות של מרווח הפעימות ומרכז הכובד של ההספק שלו מחד ובין השונות של מרווח הפעימות ומקדם התנודות המשותף, מאידך. ממצא בעל השלכות תיאורטיות מעניינות.

תרשים ה- א.ק.ג. משמש מקור מידע חשוב ומהימן על יעילות פעולת הלב. באמצעות תרשים זה ניתן לאבחן מחלות לב שונות אך מאידך ניתן גם להסיק

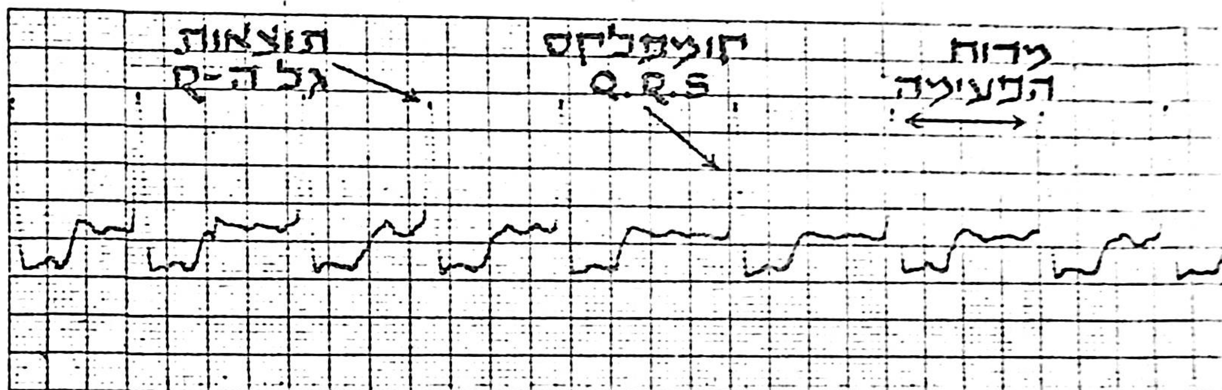
5/...

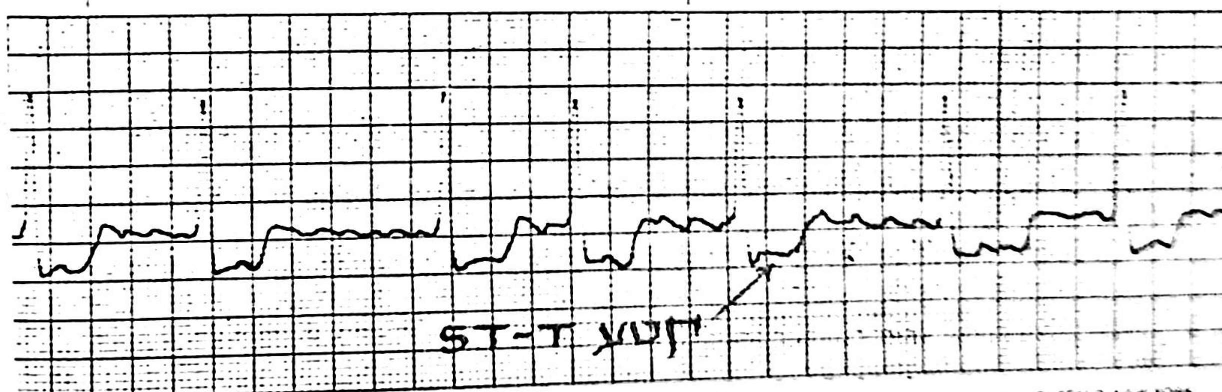
על מצבו המטבולי של הגוף כמו רמת יוני אשלגן או סידן. מסבה זו רב הפתוי להפש בתרשים סמנים המצביעים על רמת הדיגיטליס בדם. לא מצאנו התאמה בין רמת הדיגיטליס לבין מדדים שמקורם במרוח פעימות הלב. תצפית זו מצטרפת לממצאים של חוקרים אחרים אשר בדקו את הדופק הממוצע בלבד. אף הם לא מצאו קורלציה טובה. כל המדדים האחרים שקויינו ליששם לצורך חזוי רמת הדיגטליס מתנהגים בצורה דומה. בנוסף לחישובים שתוארו למעלה סקרנו את כל התפלגויות הספק האות לשנויים שניתן להבחין בהם חזותית מתוך הנחה שיתכן שלמרות חוסר הקורלציה עם השונות ניתן יהיה להבחין בשנויים בעלי משמעות בהתפלגות עצמה. אך גם בדרך זו לא נמצאו מדדים שמצביעים על רמת הדיגיטליס.

סכום

58 תרשימי א.ק.ג. הוזנו למחשב 15- PDP באמצעות מספרר קולי. מרוחי הפעימה (Cardiac interbeat interval) אשר הופקו, עובדו בשיטות סטטיסטיות ומתמטיות על מנת למצוא מדדים המצביעים על רמת הדיגיטליס בדמו של החולה.

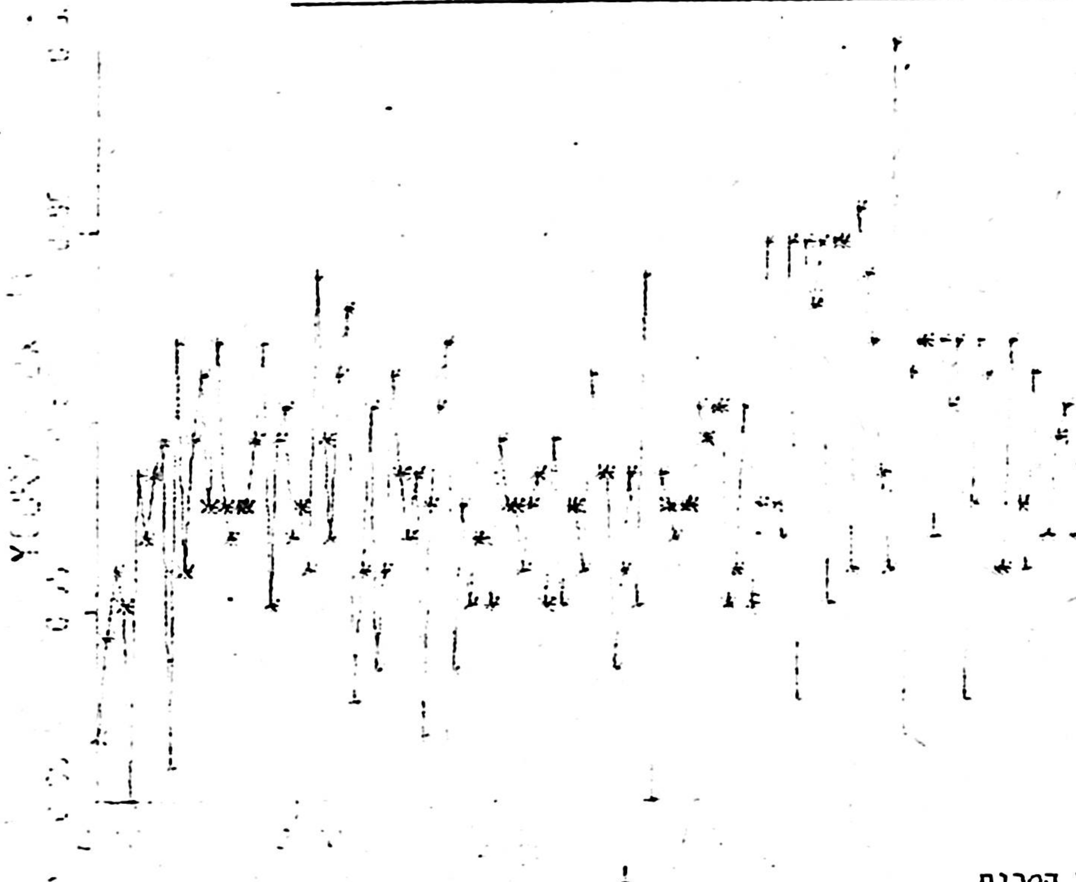
בכל מדגם מרוחי הפעימה נקבעה ההתפלגות, השונות, והתפלגות ההספק (Power Spectrum) נמצא שאין מתאם בין המדדים השונים ובין רמת הדיגיטליס בדם. נמצא מתאם גבוה בין מרכז הכובד של התפלגות ההספק ובין השונות של מרוח הפעימות.



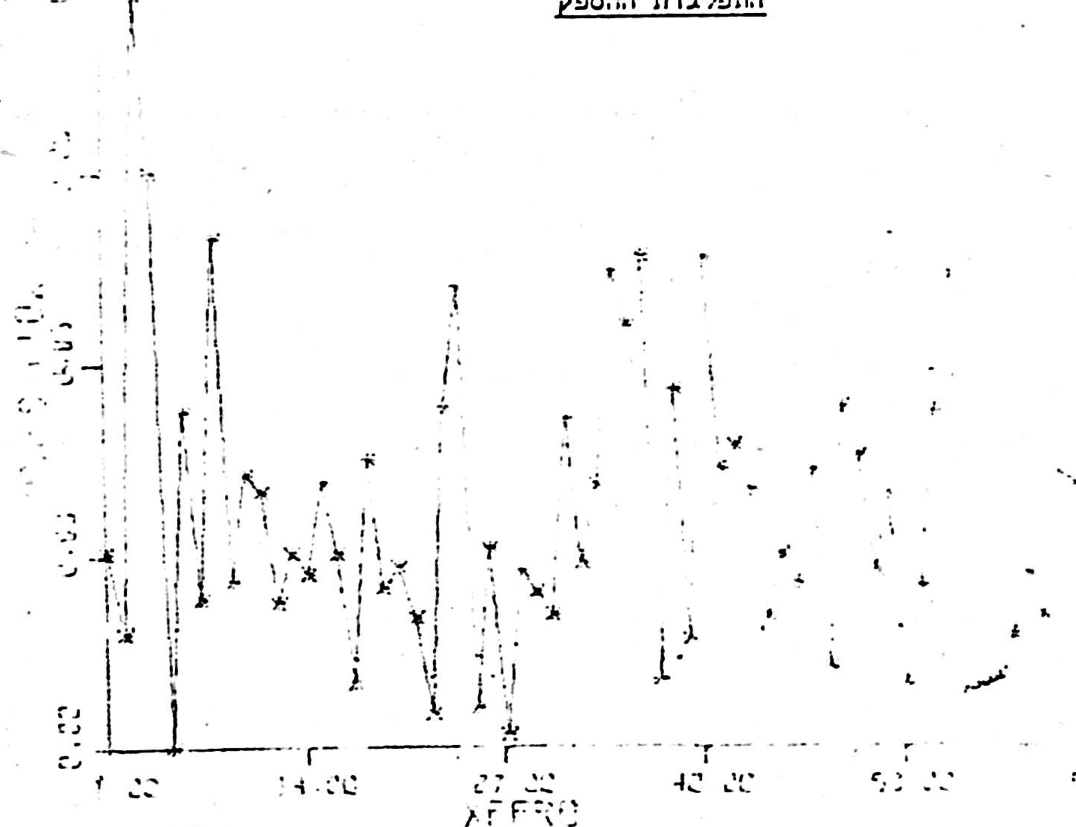


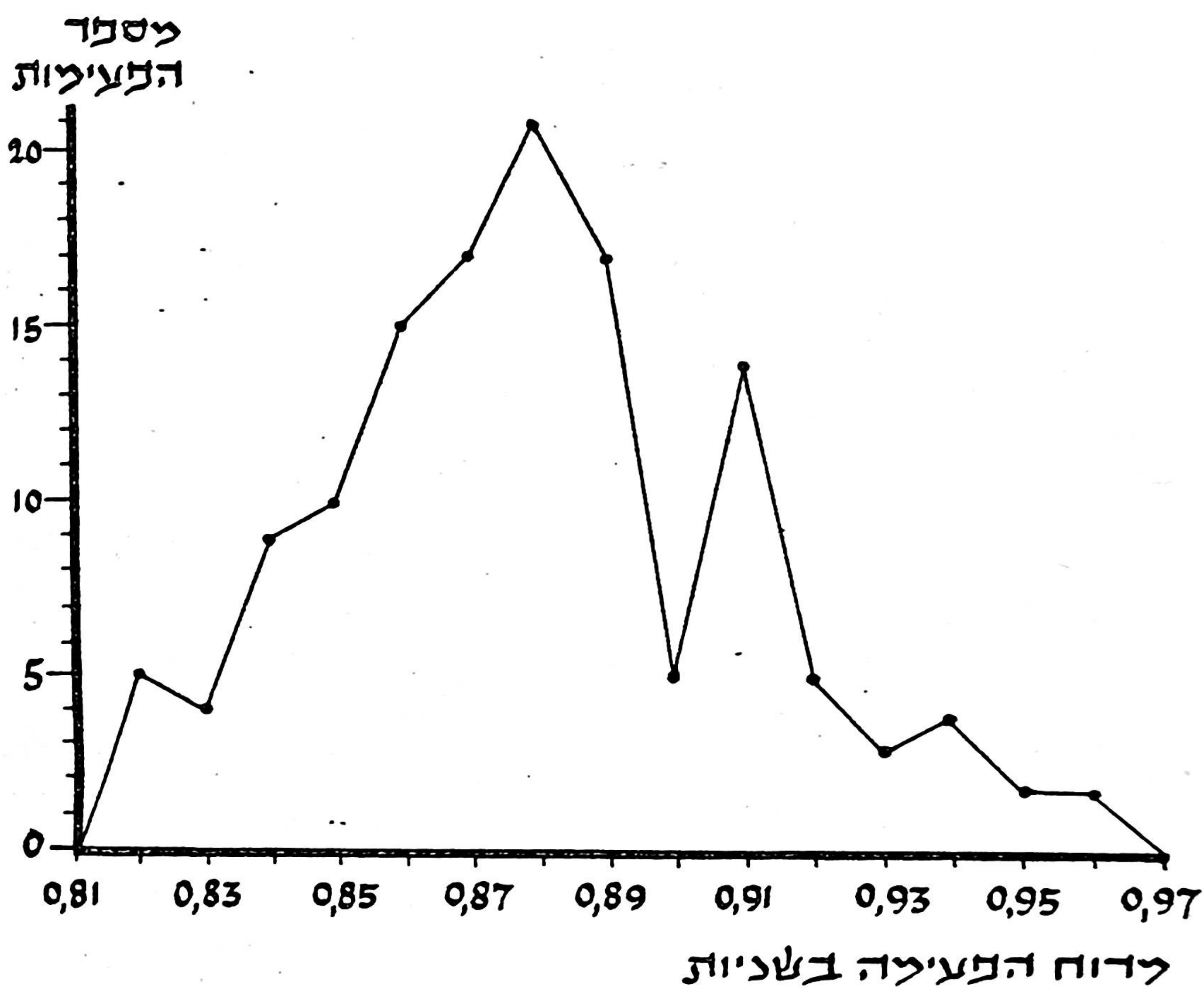
<u>טבלה מספר 1</u>. קטע מתרשים א.ק.ג. המראה פרפור פרוזדורים.

טבלה מס׳ 2   רשום מרוחי הפעימה לפי סדר הופעתם ב- א.ק.ג.



התפלגות ההספק





<u>טבלה מספר 3.</u> התפלגות מרוח הפעימה אשר נמדד ב א.ק.ג. שבטבלה מס' 1.

הפרמטר הנמדד / הערך הסטטיסטי	ממוצע	שגיאת התקן	הערך הנמוך	הערך הגבוה
רמת הדיגיטלים נגוגרם למ"ל	1.6	0.17	0.5	5.8
קצב הלב קלינית בפעימות לדקה	100	2.6	70	150
מרוח הפעימה ממוצע בשניות	0.64	0.02	0.32	0.99
מרוח הפעימות השונות	0.19	0.02	0.03	0.69
מרוח הפעימות. מקדם השונות המשותפת	0.27	0.02	0.04	0.73
מרוח הפעימות מרכז הכובד של התפלגות ההספק	0.02	0.003	0.00	0.12

טבלה מספר 4. המדדים שנצפו ב-58 חולים עם פרפור פרוזדורים מטופלים ע"י דיגיטלים.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	1					
[illegible]	0.29	1				
מידת הצניחות – [illegible]	– 0.31	– 0.72	1			
מידת הצניחות – [illegible]	– 0.31	– 0.43	0.69	1		
מידת הצניחות – [illegible]	– 0.27	– 0.39	0.61	0.92	1	
מידת הצניחות – [illegible]	– 0.32	– 0.34	0.53	0.97	0.90	1

טבלה מספר 5. ערכי מקדמי הקורלציה בין רמת הדיגטליס בדם הנבדק לבין משתנים מופקים מהתרשים האלקטרוקרדיוגרפי

"פיתוח קובץ מגדיר למעבדה קלינית

כבסיס למערכת מידע"

ש' ביילים, אמיליו קלבי, ברוך צור
חב' "יעל" לאוטומציה במינהל, ת.ד. 2848, תל-אביב

ת ק צ י ר

ערכת מידע במעבדה אמורה להשתלט על גוון רב במבנה הנתונים ובצורות השמוש בהם.

די להתגבר על קושי זה פותחה מערכת תוכנה המתחזקת "קובץ מגדיר למעבדה קלינית". ובץ זה מאורגן כעץ נתונים הכולל פריטים כגון: בדיקות מעבדה, דגימות, תהליכי עבדה, שיטות ביצוע וכיו"ב.

ישור הפריטים בעץ נעשה הן ע"י מקומם הטבעי בעץ הנתונים והן ע"י בנית מצביעים פויבטרים) מיוחדים המבטאים קשרים לוגיים מיוחדים.

ערכת ה"קומג" (קובץ מגדיר) משתתפת בתהליך פיתוח מערכת המידע במעבדה בכל שלביו:

- בשלב סיקור המצב הקיים והגדרת המערכת, אשר נעשה על ידי ראיון מנהלי המעבדה הישר לתוך טופסי ה"קומג".
- מושגת סטנדרטיזציה בשלב ניתוח השיטה, המבוסס על דוחי ה"קומג".
- התכנון המפורט מתבצע בחלקו על טופסי ה"קומג" המאפשרים קסטומריזציה של המערכת.
- בשלב התכנות במנעת ירידה לפרטים מאחר שה-Binding Time של רוב הפרמטרים נעשה מאוחר יותר, דרך הקובץ המגדיר.
- הישום והתחזוקה השוטפת מתבצעים ע"י הקמה ועדכון של הקובץ המעדכן אוטומטית את טבלאות הפיקוח של מערכת התוכנה.

מ ב ו א

בשלהי 1975 הוחלט בקופ"ח לפתח מערכת תוכנה מקומית לניהול מערכת מידע במעבדה המרכזית של קופ"ח במחוז תל-אביב, מעבדת זמנהוף.

עבודה זו הוכנה במסגרת פרוייקט פיתוח זה.

ה ב ע י ה

1.1 סיבוך מבנה הנתונים

מערכות מידע במעבדות קליניות מאופיינות, בדומה למערכות מידע ברפואה בכלל, ע"י מבנה הנתונים המסובך יחסית בעל סינטקס רופף בדרך כלל. מצב זה נובע מהכפלת כמה ממדי סיבוך:

א. סיבוך מבנה הנתונים האוביקטיבי בכל מעבדה ומעבדה, הנובע מישום תהליכים ושיטות הלקוחים משטחי מדע שונים לחלוטין (למשל בקטריולוגיה מול ביוכימיה).

ב. השונות של מערכות המידע ממעבדה אחת לשניה הנובעות משיטות עבודה ומתהליכים שונים הנבחרים ע"י הנהלות המעבדות.

ג. השנוי המתמיד בשיטות ותהליכים הנובע מהיות הרפואה בכלל נושא מחקר בעל עדיפות בשל השלכותיו הכלכליות-חברתיות והפוליטיות.

מערכת מידע במעבדה צריכה לטפל, דרך משל, בתוצאות מעבדה הבאות:

בהמטולוגיה:	תאים לבנים: 10,800 ב-CMM מהם 20% לימפוציטים.
בביוכימיה :	אשלגן : 3.1 MEQ/L חומציות: 7.48 pH
בבקטריולוגיה:	בתרבית שתן : במצאו Streptococcus Hemoliticus מעל 100,000 חידקים במ"ל.
בפרזיטולוגיה:	בבדיקת צואה: במצאו 2-10 אריטרוציטים דגברטיביים בשדה. בדיקה לביצוע תולעים: חיובי.
בבדיקת שתן :	בילירובין +++.
באימונולוגיה:	בנסיוב במצא מקטע אחיד IgG בעל שרשרות קלות מטיפוס k. יתר האימונוגלובוליבים בגדר נורמה. ההתחלקות אלקטרופורטית:

ALb	α	α_2	β	γ
56%	2.5%	6.5%	10.5%	17%+6.5%

כתוצאה מכך פותחו במקומות שונים מערכות נפרדות לענפי המעבדה השונים. כך אף במעבדה של MASS. GENERAL HOSP. המפורסם). למצב זה מגרעות משמעותיות עבור המעבדה ומשתמשיה: חשוב מאד למעבדה האימונולוגית לדעת את תוצאות הבדיקות הביוכימיות של חלבונים כאשר היא מבצעת אלקטרופרזה של חלבונים וכיו"ב.

1.2 הגוון הרב של משתמשים ושגרות העבודה

עולם המעבדות מגוון כגוון המשתמשים בהן: מעבדות בבתי-חולים ומעבדות לשרות אמבולטורי, מעבדות מחקר מול מעבדות שגרתיות, מעבדות לשרות Multiphasic Screening ואחרות. המשתמשים מכתיבים את מחזור העבודה במעבדה, את המערכת הלוגיסטית המזינה את המעבדה, את הטיפול במידע ההסטורי, את צורת התדפיסים ומבנם וכו'.

1.3 בעית הלשון והמינוח

הרפואה לשונות רבות לה: מונחים מקצועיים לטיניים בצד שמות משפחה בלשון הקודש. השילוב הדו-לשוני המקובל אף בתוך משפטי Free Text כגון "שיטת Phadebase". בעיה נוספת היא אי-האחידות של הטרמינולוגיה המקצועית; מצד אחד קיימים זה בצד זה Synonyms ומאידך קיים הרצון לטרמינולוגיה אחידה כלפי המשתמשים השונים.

1.4 בעית הכמויות המשתנות

הדרישות לבדיקות מעבדה כמוהן כחימרה - ככל שהמנהלות של שרותי בריאות מנסות להכות בהן ולהפחיתן כך הן תופחות והולכות. הגידול השנתי ב-1975 במעבדה הביוכימית המרכזית של מחוז ת"א שווה לסה"כ הבדיקות שבצעה מעבדה זו ב-1970... בצד הגידול הגלובלי יש גם שנוי ב"אופנת" הדרישה: בדיקות מסוימות מטלטות עצמן תוך כמה שנים (קולסטרול למשל) ואחרות נעלמות מן ה"תפריט". כתוצאה מכך חלים שינויים בפרמטרים כמותיים של המערכת: מהו מספר התוצאות הממוצע לדגימה? מתי להפיק דף עבודה למוקדי הביצוע? וכדומה.

2. הפתרון: "קובץ מגדיר מעבדה" (קומ"ג)

קובץ מגדיר מעבדה, כשמו כן הוא: מטרתו לאפשר ריכוז כל הנתונים והקשרים ביניהם המגדירים את המעבדה הקלינית מנקודת ראותה של מערכת מידע בצורה שתהיה עקבית ונתנת לחזרה ובצורה שתהיה קריאה וניתנת לעיבוד ע"י אדם ומכונה. מערכת התוכנה למעבדה תתאים את עצמה למעבדה ספציפית ע"י "צילומה" דרך הקובץ המגדיר את אותה מעבדה, כל שינוי בקובץ יגרור את השינוי הדרוש בהתנהגות מערכת המידע. המהר"ל מפראג היה אומר שהקובץ המגדיר הינו "השם המפורש" המוכנס מתחת ללשונו של הגולם הקרוי מערכת תוכנה למעבדה.

עולם הנתונים של מערכת מידע במעבדה מורכב מפריטים מסוגים שונים:

א. פריטים המתייחסים למושגים של העבודה המעבדתית ("דרישה", "שיטה", "תהליך מעבדה", "דגימה" וכדי) אשר הגדרותיהם ינתנו להלן.

ב. פריטים אפסנאיים המשמשים במעבדה כגון: כלי קיבול לדגימות, כלי אריזה, חומרי מעבדה, מכשירים לביצוע בדיקות וכדי.

ג. פריטים המתייחסים למרכיבים של מערך המידע עצמו - תדפיסים, טפסים וכו'.

בין הפריטים הללו קיימים קשרים לוגיים שונים. אם נתייחס לפריטים כאל צמתים ("עיגוליים") ואל הקשרים כאל קשתות ("קוים") נוכל להתבונן על "גרף מבנה הנתונים" במעבדה. נוכל להבדיל בשני סוגים של קשרים לוגיים בין הפריטים בגרף: קשרים לוגיים של שייכות וקשרים אחרים. אם נתיחס לקשרי השייכות בלבד יהיה מבנה הנתונים במעבדה "יער" של עצים, כאשר כל עץ מתייחס לרשומה בקובץ. העצים הינם מסוגים שונים - בדיקות מעבדה, דגימות, מבנה ארגוני של המעבדה וכדי.

בהתאם לשלושה סוגי הפריסים שנמנו לעיל בנוי הקובץ המגדיר משלושה קטעים. הקטע השני העוסק בפריטים אפסנאיים הריהו ככל קטלוג המשרת מערכות מלאי הספקה או תחזוקה למיניהן. הקטע השלישי כולל את הגדרת כותרות התדפיסים, מבנה טופסי הדרישה, פרמטרים מוכנים למערכת Data Entry וכדומה; בהמשך סעיף זה נדון בעיקר בקטע הראשון המכיל את יער מבנה הנתונים במעבדה. לשם כך דרושות לנו מספר הגדרות:

<u>בדיקה</u> - תהליך מעבדתי שמטרתו בירור פרמטר בריאותי מסוים.

<u>דגימה</u> - דוגמא מחומר בגוף הנבדק, הנלקחת בנקודת זמן ומקום אחת.

<u>דגימה מקובצת</u> - צרוף של כמה דגימות.

<u>דרישה</u> - הגדרת התהליכים שעל המעבדה לבצע. יחידת הדרישה הקטנה ביותר היא הצמד בדיקה-דגימה או בדיקה-דגימה מקובצת.

<u>דרישה מקובצת</u> ("פרופיל") - אוסף של צמדים בדיקה-דגימה אשר הרופא יכול לדרוש את ביצועם המשותף תחת שם אחד.

<u>שיטת ביצוע הדרישה</u> - המתודה המעבדתית לפיה מבוצעת הדרישה, אותה דרישה ביתן לבצע בכמה שיטות אופציונאליות (דוגמא - בדיקת המוגלובין בקולטר ובדיקה בשיטה ידנית).

<u>תהליך מעבדה</u> (או תת-בדיקה)- תהליך מעבדתי יחיד המתבצע בתחנת עבודה אחת. דרישה כוללת לפחות תהליך מעבדה אחד.

<u>שיטת ביצוע תהליך מעבדה</u> - מתודה מעבדתית אשר לפיה מבוצע תהליך מעבדה.

<u>נתון מקובץ</u> (או מדידה) - קבוצת נתונים בעלי מכנה משותף וקשר סינטקטי וסמנטי, הנשלחת תמיד לרופא כיחידה אחת. זוהי יחידת המידע הבסיסית ברמה של הרופא.

<u>נתון</u> - יחידת המידע הבסיסית במעבדה, הנתון נרשם ע"י עובד המעבדה בנקודת זמן אחת או מחושב כפונקציה חד-ממדית על בסיס נתונים אחרים.

מהם יחסי הגומלין בין ההגדרות לעיל?

אוסף דרישות הוא המרכיב הבסיסי של המערכת. אלה הם השרותים אותם מסוגלת המעבדה להציע למשתמשיה.

לנוחיות המשתמשים מאפשרת המערכת הגדרה של דרישות מקובצות, היינו - מתן שם מקבץ לכמה דרישות לשם הכנסת נוהלי דרישה אחידים במוקדי שרות רפואיים מסוימים או לשם הקלה על הרופאים (ועל המערכת) בדרישה של פרופילים מקובלים. (דוגמאות: ביתן להגדיר "בדיקות חלבונים" כקבוצה הכוללת: סה"כ חלבונים, אלבומין, גלובולין ואולי גם β-Lipoproteins. ביתן להגדיר את "בדיקת קבלה למחלקה פנימית ג'" אשר תכלול המוגלובין, לויקוציטים, סוכר, אוריאה וקולסטרול. דוגמא נוספת - ביתן להגדיר את אוסף הבדיקות המבוצעות עבור מכוני בדיקה המונית (AMHT) תחת שם אחד).

מנקודת מבט של חלוקת עומס ביצוע עבודה, הרי למנהלת המעבדה ולמערכת התוכנה יש חשיבות בפיצול הדרישה לתהליכים אחדים במבוצעים בתחנות עבודה שונות (יתכן, אך לא בהכרח שתהליכים אלה מבוצעים בימים שונים, ע"י עובדים שונים). תהליכים אלה המרכיבים את הבדיקה נקראים כאמור תת-בדיקות או תהליכי-מעבדה. (דוגמאות: בדיקה מקרוסקופית היא תת-בדיקה של דרישת תרבית צואה במעבדה בקטריולוגית).

במוקד ביצוע התהליך המעבדתי נעשות עבור תת-בדיקה כלשהי, מדידה אחת או יותר. מדידות אלו הן שנשלחות בסופו של דבר אל הרופא. (למשל, מדידת לויקוציטים בתת-בדיקה מקרוסקופית של בדיקת שתן במעבדת דיאגנוסטיקה כללית). מדידה כזו מתקבלת מריכוז

נתון אחד או יותר. (דוגמא: מספר הלניקוציטים וצורת הופעתם בקבוצות או כבודדים, הם שני נתונים של מדידת לויקוציטים).

בלוח 2.1 מוצגת הסכמה של עץ מבנה נתוני המעבדה, דוגמא מפורטת המתארת הצבה של בדיקת שתן בסכמה זו מוצגת בלוח 2.2.

3. מבנה הקובץ המגדיר

הערה: בסעיף זה נדון בקטע הראשון של הקובץ בלבד.

3.1 אבני בנין

נציג תחילה מספר מושגים בסיסיים:

ערך: שרשרת של תויות (למשל '+', 'רבינוביץ', '17.6').

שדה: פונקציה, המצמידה לאוסף ערכים מסוים משמעות כלשהי אם הם מופיעים במיקום מוגדר. בדרך כלל מתיחס המונח שדה למיקום בו מופיעים ערכים אלה.

נגררת: קבוצת שדות בעלי מכנה משותף, המתיחסים לאותו פריט. בדרך כלל המכנה המשותף הוא "אחדות הזמן והמקום" של היווצרות מופעי השדות. מקובל גם שנגררת נוצרת תמיד מטופס מקור אחד (לעיתים כמה נגררות על אותו טופס). הנגררת היא יחידת הגישה הישירה הנמוכה ביותר בקובץ.

רשומת פריט: זהו אוסף של נגררות המתיחסות לאותו פריט במערכת המידע.

עץ בדיקה: זהו אוסף נגררות המתיחסות לאוסף של פריטים אשר בין כולם קיים יחס שייכות, ואשר אין להם יחס שיכות עם פריטים אחרים בקובץ.

מופע: (שדה, נגררת, רשומת פריט, עץ בדיקה, קובץ) הינו מקרה פרטי של ההגדרה (של שדה, נגררת וגו').

הקטע הראשון של הקובץ המגדיר הוא אוסף כל מופעי עצי הבדיקות המוכרות במעבדה.

3.2 מבנה עץ הבדיקה בקובץ

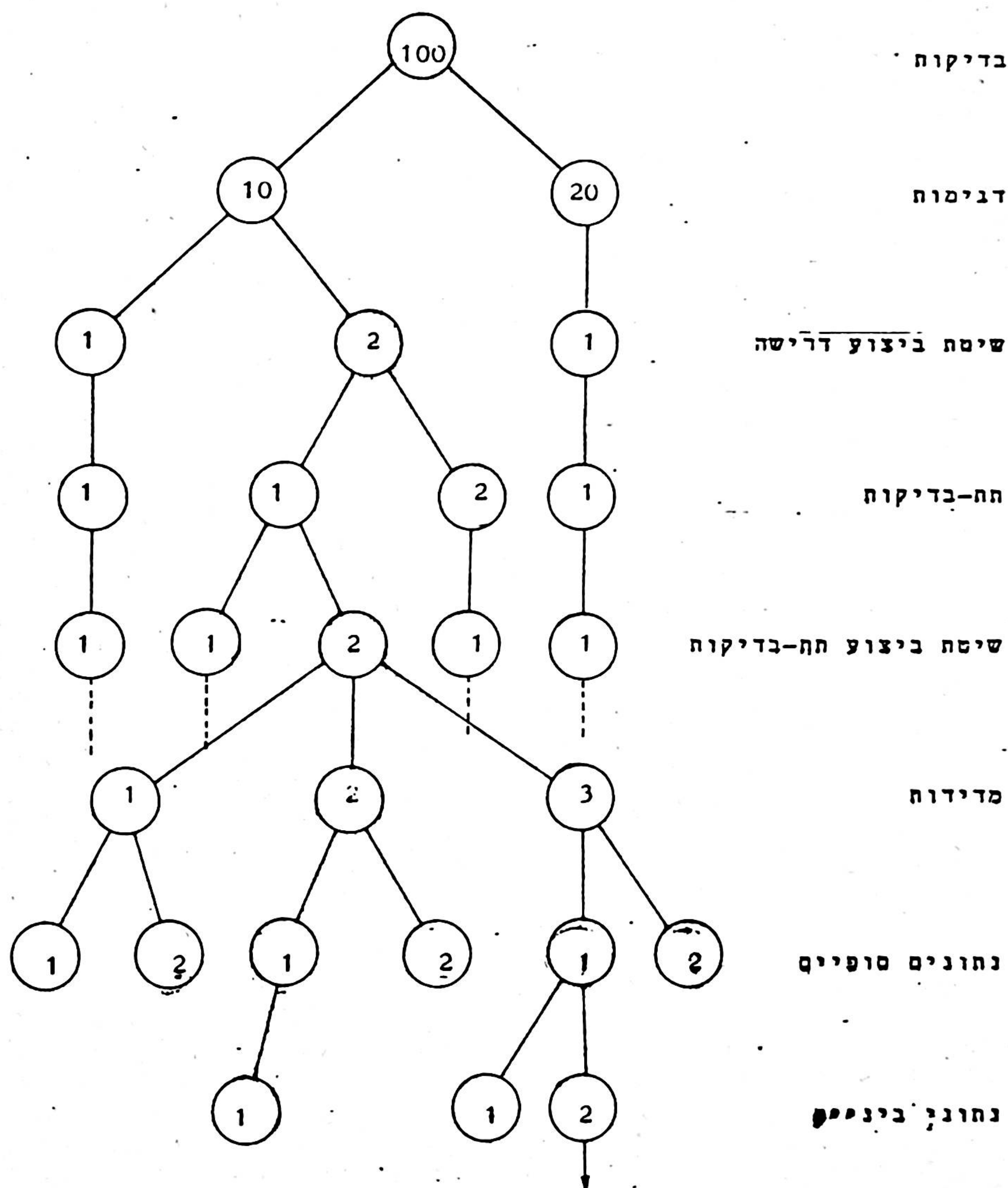
לכל נגררת בקובץ ספח זהוי הממקם אותה נכונה בעץ הבדיקה אליו היא שייכת, ראה לוח 3.3.1. עץ הבדיקה יקבע לפי "המספר הקטלוגי" של הדרישה. הנגררת שייכת לפריט (בתוך עץ הבדיקה המסוים) אשר זהוין נתון ע"י "מפתח הפריט". כיצד נוצר מפתח הפריט? - ע"י שרשור המספרים הסודרים של כל הקשתות במסלול המוליך משורש העץ לפריט. (ראה שוב לוח 2.1). קוד הנגררת מאפשר לצרף מספר רב של נגררות לאותו פריט.

במפתח הפריט מופיעים אפסים בכל שדות הזהוי המתיחסים לרמות נמוכות מן הפריט הנוכחי בעץ.

לכוכביות המופיעות במפתח הזהוי משמעות מיוחדת: אלו תויות אשר כל השוואה שלהן לתוית אחרת נותנת תוצאה חיובית (לכן ה-* נקראת בסלנג הפרוייקט - "ג'וקרי").

ניתן להכניס לקטלוג נגררות עם כוכביות בשדות הזהוי על מנת לאחד צמתים בעץ, בלי הגבלה של פירוק הענפים חזרה ברמה נמוכה יותר בעץ; לדוגמא: במקרה של שתי שיטות ביצוע לאותה דרישה, הזהות בכל פרטיהן, פרט, נאמר, לערכי היחוס של תוצאות שתי השיטות. במקרה כזה נכניס בכל הנגררות המתארות את מבנה הנתונים * בשדה הזהוי של "השיטה" פרט לנגררות ערכי היחוס בהן נפרט שיטה '1' ושיטה '2'.

לוח 2.1 : סכמה של עץ נתוני בדיקת במעבדה



לוח 2.2 : עץ נתוני בדיקת שתן

כלומר, ניתן לומר שהתיחסות לצומת עם מילוי אפסים לרמות הזהוי שמתחתיה משמעה המשמעות לצומת זו עצמה ולה בלבד. מילוי ג'וקרים לרמות הזהוי הנמוכים משמעו המשמעות לכל תת-העץ שהצומת דנן מהווה שורשו.

3.3 הנגררות המשתתפות בבנין עץ הבדיקה

להלן תאור תמציתי של הנגררות העיקריות:

נגררת שמות (A)

ניתן להצמיד לכל פריט שמות מהסוגים הבאים:

א. שם מלא (50 תויות)
ב. שם מקוצר (14 תויות)
ג. שם מקודד (4 תויות)

לכל פריט יכולים להיות כמה שמות מכל סוג, כאשר השם הראשון הינו השם המקובל בטרמינולוגיה הסטנדרטית והאחרים הם Synonyms.
שפת הדווח: ניתן למלא שמות באותיות עבריות או לטיניות. ניתן אף לערב את שני סוגי האותיות בתוך אותו שם, כאשר הטקסטים בעברית מסוגרים בסוגריים זויתיים (< >).

נגררת דרישה (D for Demand)

נגררת זו מתיחסת לצעדים בדיקה × דגימה וכוללת פרטים כגון: סוג הדגימה, המידע הדרוש למעבדה לביצוע הדרישה, זמן תגובה בדרש של המעבדה (מינימום ומקסימום), פרטי מחיר וכדי.

נגררת שיטה (M for Method)

נגררת זו מתארת את פרטי תהליך העבודה של השיטה המעבדתית. היא כוללת פרטים על' אמצעי זהוי נדרשים, פעולות קליטת הדגימה, תחנת הביצוע, מיכשור לביצוע, תבנית הביצוע במעבדה (מתי להוציא דפי עבודה, גודל אגיד הביצוע) וכיו"ב.

נגררות הנתון

נגררת סוג הנתון (א)

זו מזהה את סוג הנתון מבחינת הפורמט שלו, ומופעיו בדוחות המעבדה השונים (דפי עבודה, תדפיסי סיכום, תוצאות לרופא).
תיאור הנתון נבחר כאחד מ-5 אפשרויות: נתון נומרי, נתון סמי-כמותי (היכול להיות בטווח של י-י עד יאאאי, או בפורמט של "חיובי-שלילי", בתון שהוא צירוף של מספר מקודים מתוך רשימה של אפשרויות חשובה, וכן נתונים שהם טקסטים חופשיים.

נגררת נתונים נומריים (ס)

כאן מופיעה של יחידות המידה, גבולות סבירות, וכל הפרטים הקשורים לנוסחאת החישוב אם הנתון מחושב.

נגררת אפשרויות תשובה (P for Possible Texts)

בה מקום לתשובות בפורמט זהה לנגררת השמות (על שלושת סוגי הפירוט: טקסט מלא, מקוצר ומקודד) ובנוסף להם שדות פיקוח המאפשרים ביטוי של יחסי גומלין בין אפשרויות התשובה השונות.

נגררת דגימות מיוחדות (Q for Quality Control)

מטרתה להזין למערכת נתונים הדרושים לקליטת תוצאות של Quality Control Sera ושל דגימות Standard ו- Blanks.

נגררת ערכי יחוס (R for Reference Values)

כוללת מספר צמדים של גבול עליון וגבול תחתון של ערכי יחוס המתיחסים לאוכלוס-יות שונות (עד 100 קבוצות אוכלוסיה).

נגררת הדגימה (S for Specimen)

בה כל פרטי הדגימה הכוללים את תאור החומר הפיסיולוגי; שיטת האריזה והמשלוח, תנאי לקיחה, חומרים מוספים וכדי. ניתן להגדיר גם דגימות מקובצות ע"י קישור בעץ דגימה של מספר נגררות דגימה. למשל עבור בדיקת Creatinin Clearance דרושה דגימה הכוללת דגימת דם, דגימת שתן אחת לאחר שעה ודגימת שתן נוספת לאחר שעה נוספת.

נגררת קישור (X for Cross)

נגררת זו מיועדת לקישור הפריט אליו היא שייכת לפריטים אחרים. קישור זה יכול להיות בעל משמעויות שונות כגון: אחוד פריט זה עם פריט בקובץ; העתקת נתובי פריט זה מנתוני פריט אחר, הכללת פריט אחר בדווח רופא במקום פריט זה וכיו"ב. לכל שדה קישור קודם שדה התניה המאפשר לבנות קישור מותנה כגון: אם בבדיקה איכותית של סוכר בשתן התגלה סוכר צפה לביצוע בדיקה כמותית של סוכר בשתן. ההתניה יכולה להיות תנאי נומרי או תנאי כגון פתולוגיות, חריגה מערכי יחוס וכדי.

נגררת הערות (Z)

מאפשרת "השחלת" Free Text בצד כל נתון בקובץ. הערות אלו מופיעות בתדפיס הקובץ המשמש כספר הדרכה לישום המערכת וניתן יהיה לשלוף אותם גם דרך מסכי המשתמשים.

בלוח 3.3.1 מצורך כדוגמא הטופס המשמש לאסוף נתובי נגררת סווג נתון (N).

4. השמוש בקובץ הגדרת המערכת בשלבי הפיתוח

4.1 סיקור המצב הקיים להגדרת המערכת

ריכוז המידע על מערכת המידע הידבית אשר מתופעלת במעבדה נעשה על ידי צוות הפרוייקט תוך קיום פגישות עם מנהלי המחלקות במעבדה המרכזית של מחוז ת"א (מעבדת זמנהוף). לצורך זה הוכנו טפסים שהותאמו למילוי על ידי המזכירה הרפואית שבצוות. עיבוד הנתונים נעשה Off-Line, ובשלב ראשון הופקו דוחות חוזרים למנהלי

N

DATA ITEM FORMAT

Major Key				Item Key						Attribute Key			Op. Code
Subject	Test N.O.	Specimen	Check D1	Meth 1	Proced.	Meth 2	Measur.	Datum	Check D2	Form N	Seq. No.	Cont. 1	Create; 1 Update; 2 Indirect U/D; 3 Cancel; 9 Indirect Canc
(1)	(2-7)			(8-13)						(14-17)			(18)

45. Datum Type:

By Stage	Final Intermediate	F I	19
By Usage on Lab Printouts	Bench Work Sheets Yes, Appears always with 'x' No, Does not Appear Yes, without 'x' (draft, notfeeded Yes, Always with 'Ø' (optional, .technitian decides)	X N E Ø	20
	Verification WorkSheet — Yes / No		21
	On Lab-Summary Rep. — Yes / No		22
By Computation	Computed Observation		23
Reporting	Appears on Doctors Report — Yes / No By Condition on Itself(*) By Condition Depending on Other Datum		24
	Indication for out of Ref. Val. or Pathology Results. — Yes / No		25
	Partial Reporting is Done after Wait of Maximum:		26-28

(*) Use Condition on Linkage Form. (Link Code 8)

46. Format

Numeric	1
Free Text	4
Semi-Quant. (+)	2
Coded Text	5
Positive-Negative	3
	29

Numeric:

Position for Fraction	30
Position for Integers	31
10th Exponent	32

→ Max. Length (30 32)

Semi-Quant.:

Max. + Degrees	30
Max. − Degrees	31
Existance of +	32

→ No. of Length (30-31); Max. No. Required (32)

Positive-Negative:

Exist. of Weak Pos.	30
Exist. of Strong Pos.	31
Exist. of Very Strong Pos.	32

4.3 שלב התכנון המפורט

בשלב זה הוכנסו מספר שינויים ותוספות לקובץ, על מנת לאפשר קליטת הפרטים הנוספים שעל מערכת התוכנה להיות מוזנת בהם. החלסות שונות שהתקבלו הוזנו דרך הטפסים לקובץ, על מנת לייצג את המעבדה" לדמותה המתוכננת.

4.4 שלב התכנות

הסיבוך הרב של מבנה הנתונים גובש לסכמות ניהול נתונים אשר תוכנתו. התנהגותה המפורטת של מערכת התוכנה נקבעת למעשה ע"י הפרמטרים שבקובץ המגדיר. עם זאת לא היה להערכתנו חסכון ממשי בשלב התכנות בשל דרגת ההפשטה הגבוהה יותר של התכניות.

4.5 ישום והרצה

שלב זה עדיין לא בוצע. יהיה צורך ב"קיזוז" המערכת לתפעול לפי בקשות צוות המעבדה. בשלב זה יוחלט, בשיתוף המעבדה, אילו בדיקות תופענה על איזה דפי עבודה (ובכך יגררו עמם את כל מבנה הנתונים המוגדר ברמות הנמוכות יותר של העץ למבנה דף העבודה), יוחלט מתי להוציא דפי עבודה שונים, וכדי. הנתונים האלה יועלו בקטע השלישי של הקובץ המאפשר פרמטריזציה של מערכת הדווח.

4.6 תחזוקה

כל שנוי אשר יתבקש במבנה המעבדה, ספקטרום השרותים שלה (כולל הגדרת "פרופילים" הניתנים לדרישה ע"י הרופאים), שיטות ביצוע בדיקות, מבנה מערכת הדווח, הגדרת אוכלוסיות מעקב נוספות לשליפה ועיבוד סטטיסטי וכדי, יוכל להיות מתורגם לתוכנה ע"י מערכת הקובץ המגדיר.

4.7 תיעוד המעבדה

מצבור הפרטים הגדול בקובץ והאפשרויות לצרף אליו פרטים נוספים ע"י הוספת סוגי נגררות חדשות וכן הערות והסברים בפורמט חופשי הביא אותנו להפקת Manuals של המעבדה בכמה חתכים כדלקמן:

א. ספר עובד מעבדה כולל את רוב פרטי הקובץ הנוגעים למערכת ביצוע הבדיקות. ראה דוגמא קצרה בלוח 4.7.1.
ב. רשימה לאחיות הממונות על אסוף הדגימות ובה פרטים על הדגימות הדרושות לכל בדיקה והנחיות לתנאי לקיחתן, אריזתן ומשלוחן וכן רשימת פריטי המידע הדרושים במעבדה לצורך ביצוע הבדיקות.
ג. תדפיס לרופאים. בתדפיס הזה יוכנסו הפרטים בעלי המשמעות לעבודת הרופא כגון: ערכי-יחוס, הנחיות למילוי טופס דרישת הבדיקות, הנחיות שיש למסור לחולה בנוגע לתנאי מסירת הדגימה, משך הזמן הממוצע בו אמורה המעבדה לבצע את הבדיקה וכדי.

5. השימוש בקובץ המגדיר להנחית מערכת התוכנה

[מע]רכת התוכנה אשר תופעל במעבדת זמנהוף מבוססת על מחשב Mini. שמוש בקובץ המגדיר [במ]לואו ובמבנהו המסובך הוא מעמסה כבדה יחסית לתוכנה המפותחת למחשב בסדר גודל זה. [לכ]ן יבוצע תמצות של נתוני הקובץ לשתי טבלאות פיקות ולמילון.

... טבלת הדרישות - בטבלה זו כל המידע הדרוש לקליטת הדרישות לבדיקות מעבדה ולניהול הקבצים הכרוך בכך: ההנחיות המלאות להפקת מדבקות לזהוי הדגימות במעבדה, הנחיות לגבי הרכב הנתונים של כל בדיקה, הנחיות לגבי הכללת הבדיקות בדפי העבודה למיניהם וכדי. טבלה זו תשב בזכרון בזמן פעולת המערכת.

[illegible] ופוספות. פותחה תכניה לניהול הקובץ בה יושמו כל התכונות [illegible]

א. [illegible] לכל רשומת קישור, כך שכל קשר הנבנה במערכת הוא דו-צדדי. [illegible], ביטול או שינוי בזהוי של רשומת קישור משפיע לכן גם על [illegible].

ב. [illegible] של עדכון של קבוצות של רשומות פריט כולל שינוי של פרטי הזהוי [illegible] (על מנת לאפשר להזיז ענפים מעץ לעץ). עדכון כזה קרוי בסלנג [illegible]".

ג. [illegible] במידה זו או אחרת לכמה עצים. במערכת נשמרת האפשרות [illegible] בכמה רמות עומק:

[illegible] 1 [illegible] של פריט מסויים משתתף בכמה עצים. (למשל כאשר בתוך ששמו "כמות [illegible] ל-24 ש'" משתתף בחישוב התוצאה הסופית של כמה בדיקות המבוצעות [illegible]) במקרה של מאוחדים הצמתים המתאימים בעץ ע"י שמוש בנגררת [illegible] קישור E for Equivalence.

[illegible] 2 כאשר ההגדרה של פריט מסויים זהה להגדרה של פריט אחר במערכת (למשל [illegible] בכמה בדיקות בקטריולוגיות). על מנת לחסוך במקום, [illegible] בנגררת קישור עם קוד קישור C for Copy, במקרה זה כל שינוי [illegible] המקושר משנה גם את הפריט המצביע אליו.

[illegible] 3 כאשר ההגדרות של פריט מסויים דומות להגדרה של פריט אחר, אולם [illegible] שינויים שיבדילו ביניהן. במקרה זה מאפשרת התוכנית [illegible] בנגררות על ידי הכפלתן בקובץ פשוט כמשמעו ע"י תנועת העתקה [illegible]. העתקה זו ניתן לבצע על קבוצה של נגררות ע"י שמוש [illegible] - ראה סעיף 3.2).

[illegible]

[illegible] במידה רבה על תדפיסים שובים אשר הופקו מן הקובץ. לצורך [illegible] מיוחדת, אשר אפשרה, למעשה, שליפת הנתונים בכל מבנה [illegible] (4.7.1).

[illegible] ניתן לקבל את הנתונים הבאים:

א. [illegible] של בדיקות שמתבצעות במעבדה.

ב. [illegible] המתנהלת במעבדה והבדיקות השיכות לכל ספר רישום (הגדרת [illegible]).

ג. [illegible] של עובדי המעבדה לדעת בבצעם את הבדיקות.

ד. [illegible] קודמות לאותו נבדק בכל בדיקה ובדיקה.

ה. [illegible] בשמוש המעבדה, ואילו בדיקות מופיעות על כל אחד מהם.

ו. [illegible] אשר בשמוש המעבדה ובאילו בדיקות זקוקים להם.

ז. [illegible] של כל בדיקה ובדיקה (באיזה ימים ובאילו שעות).

ח. [illegible] אשר בשמוש המעבדה ואילו בדיקות מבוצעות עליו.

ט. [illegible].

י. [illegible] האחסון וזמני ההשהיה המותרים לגבי כל בדיקה [illegible].

```
SYDEF KEY      UPDATED|
C.35C.33.8   05/08/75|
                      |
ITEM'S KEY            |
C.C.C.0.0.A.0C.2      |*NAME CF TEST
                      |
                      |  FULL NAME= TRANSAMINASE  GPT                 -SHORTENED NAME=TRANS. GPT.   CODE=GPT .
                      |
                      |  SYNONIMS = ALANINE  TRANSAMINASE
                      |
0.C.0.0.0.0.C0.7      |*SPECIMEN  CHARACTERISTICS
                      |
                      |      *|NAME                                   *|QUANTITY REQUESTED |*MINIMAL QUANTITY
                      |       |                                        |                   |
                      |       |BLCOD                                   |C.C.  5.0          | C.C.  3.5
                      |
                      |*TEST REQUISITICN
                      |
                      |  TEST'S NAME APPEARS EXPLICITELY CN BIOCHEMISTRY REQUEST FORM
                      |
                      |  FOLLOWING INFORMATICN IS RELEVANT FOR TEXT EXECUTION AND INTERPRETATION=
                      |
                      |  -SEX
                      |
                      |  -DIAGNOSIS
                      |
                      |  -DRUGS
                      |
                      |*RESULTS STORAGE
                      |
                      |  STORAGE CF RESULTS OF PRECEDENT TESTS= NOT ESSENTIAL
                      |
                      |*RESULTS REGISTRATICN
                      |
                      |   ON BIOCHEMISTRY MASTER  WCRK SHEET
                      |
                      |*MINIMAL RESPONSE TIME
                      |
                      |   1 DAY
                      |
                      |*MAXIMUM ALLOWED RESPONSE TIME
                      |
                      |   5 DAYS
                      |
                      |*AVERAGE NUMBER OF TESTS REQUESTED IN A DAY
                      |
                      |   10 TESTS
```

ב. טבלת דפי העבודה - כל הנתונים הדרושים לניהול דף עבודה (הדפסתו, קליטת תוצאות עבורו, ביצוע חישובים, וכדי) ירוכזו בנפרד לכל סוג דף עבודה. כאשר תצטרך המערכת לבצע פעולה כלשהי על דף עבודה ישלפו הנתונים הדרושים לזכרון וישארו שם עד תום העיבוד.

ג. המילון - במילון ירוכזו שמות וקודים של כל הפריטים במערכת, בעיקר לשמוש בזמן הדפסת התשובות לרופאים. הגישה למילון היא דו-כיוונית - הן דרך המפתח הנומרי של הפריט, והן דרך הקוד האלפא-נומרי שלו, וזאת כדי לאפשר לעובדי המעבדה ליישוחח" דרך הצגים בלשון הברות הברורה להם.
לדוגמא: הזנת שמו של אורגניזם במעבדה הבקטריולוגית אשר איננו נמצא ברשימת האורגניזמים הנפוצים המפוענחת על דף העבודה.
ע"י החזקת "משקל-מיון" לכל אחד מהטקסטים במילון ניתן יהיה ל"סדר" את הדוח לרופא לפי כל סדר רצוי. (למשל - לסדר את הבדיקות המודפסות לפי מערכות פיזיו-לוגיות הנבחנות דרכן).

סיכומו של דבר, אפשרה "תפיסת הקובץ המגדירי" השתלטות על עולם המידע המעבדתי בצורה יסודית ועקבית, איסוף המידע המפרט על כל משך הפרוייקט להקלת הלחץ הכבד של שלב הישום. יתר על כן, על בסיס גישה זו נוח היה לפתח את מערכת התוכניות כתבילת תוכנה מודולרית וגמישה הניתנת לרפליקציה במעבדות רבות.

ת ו ד ה

עבודה זו לא היתה מונחת בפני חקורא ללא העזרה השופעת והיעוץ המקצועי של צוות המעבדה המרכזית של קופ"ח בתל-אביב.

תודתנו נתונה לד"ר מירה להב ולכל צוותה: החל במנהלי המחלקות וכלה באחרון העובדים.

תודה מיוחדת מוקדשת לד"ר פנחס ירדן, האחראי על הפרוייקט מטעם מרכז קופ"ח אשר לווה אותנו לווי צמוד לאורך כל הדרך.

קבוצת דיון מס' 5:

"יישומים ב- APL כנגד שפות עיליות אחרות"

יו"ר: ד"ר ד. כהן

צ. וייס

מ. שניידר

י. ונד

א. סגל

APL - היבט כללי על השפה ושימושיה

צבי וייס
מרכז מדעי י.ב.מ., ישראל
קרית הטכניון, חיפה

מבוא

בשנים האחרונות אנו עדים לתופעה של עליה מהירה במדת הפופולריות של השפה APL. התפשטותה של השפה מתבטאת במספר הגדל והולך של יצרני מחשב המשווקים מערכות APL, במספר המוסדות האקדמיים ובתי הספר המלמדים את השפה, בכמות המאמרים והספרים אשר נושאם קשור לשפה, וכמובן בגדול הנכר בנצול השפה כאמצעי לפתוח תכנה בתחומים רבים של שמושי המחשב. במקביל להתפתחויות אלה וכתוצאה מן הנסיון שנרכש תוך כדי נצול השפה התגבשו בקרב אנשי המקצוע דעות באשר לתועלתה ומדת מעשיותה של השפה. הדעות נחלקות בין מצדדי השפה, הרואים בה אמצעי רב תכליתי, גמיש ובעל עצמה מרובה, לבין מתנגדיה אשר עקר טענותיהם מתרכזות באי יעילותה של השפה ובקושי לקרוא ולהבין תכניות הכתובות ב-APL.

סקירה זו של APL מטרתה להציג בקצרה את השפה ולהדגים, ולו במעט, את אפייה המיוחד. הקורא אשר יגלה ענין נוסף בשפה מופנה לרשימת המקורות המצורפת למאמר.

רקע היסטורי

ראשי התבות APL (A Programming Language) נטבעו ע"י K.E. Iverson עם פרסום ספר, הנושא שם זה, בשנת 1962 [1]. בספרו מציע Iverson שפה מדויקת, תמציתית, ובעלת אופי של נסוח מתמטי לצורך הצגת פתרונות אלגוריתמיים בשטחים שונים בתחום עבוד הנתונים.

נסיונות ראשונים למִמוש השפה על מערכת מחשב נעשו בעקר במרכז המחקר של י.ב.מ. ב- New York, Yorktown Heights. מערכות אלה מתוארות ע"י H. Hellerman [2], Falkoff and Iverson [3], Breed and Lathwell [4]. בשנת 1968 הוכרזה מערכת APL\360 [5] ונתנה לראשונה האפשרות לרכוש מערכת APL מאת יצרן מחשבים.
מערכת APL\360 מממשת חלק מהשפה המקורית המוגדרת בספרו של Iverson. המערכת תוכננה ונכתבה כמערכת אינטראקטיבית הנותנת אפשרות לתכנות ושמוש on-line. מרכיבי המערכת הנם תכנית מפקחת ומפענח, התכנית המפקחת מטפלת בתקשורת עם המסופים, בנהול שתוף הזמנים, בתרגום תכניות מצורת הצגתן החיצונית להצגה פנימית ובנהול ספריית התכניות, בעוד המפענח מבצע את התכניות עצמן.

יצרני מחשב נוספים הבחינו במדת העִנְין אשר מעוררת השפה ובפוטנציאל המכירות שלה, בעקבות הכרה זו פותחו מערכות APL על מחשבי CDC ,Burroughs ,Univac,

DEC, Xerox, Siemens ועוד.

מערכת APL\360 זכתה אמנם לתפוצה נכרת, אולם החלטות שונות שנעשו בשלבי-הפתוח הראשונים של המערכת הגבילו את תחום שימושיה ליישומים המסתפקים בכמות מצומצמת של זכרון, אינם מתבססים על קבצים ונתן לענות על דרישות הקלט/פלט שלהם באמצעות מסוף איטי. מגבלות אלו הוסרו עם פתוחן של מערכות APL נוספות, דוגמת APL PLUS [6], APLSV [7], APL/700 [8]. מערכות אלו מספקות, באופנים שונים, אמצעים לעבוד קבצים ולהכוונת קלט/פלט להתקני קלט/פלט מהירים (קוראי כרטיסים ומדפסות). הופעתן של מערכות חדישות אלו גרמה, כצפוי, לשנוי במרכז הכובד של שמושי APL. השפה אינה נחשבת יותר לשפה המתאימה בלעדית ליישומים "מדעיים", אלא נמנית בין השפות הרב-תכליתיות ומשמשת לפתוח תכניות במגוון רחב של נושאים. יחד עם זאת נקדים ונאמר כי APL מנוצלת בעקר ביישומים בהם גמישות התכניות, משך זמן הפתוח ודרישה לבצוע שנויים ועדכונים תכופים בתכנה הנם גורמים בעלי חשיבות עיקרית. ממוש מערכות APL נעשה באמצעות תכנית מפענחת (interpreter), עובדה המתבטאת במחיר גבוה, יחסית, להרצת תכניות הכתובות בשפה זו.

מבני נתונים

הרומתה העקרית של APL לתחום שפות התכנות מתבטאת בשלוב הנובע מאמוץ המערך כמבנה הנתונים היחיד של השפה ומהגדרה של אוסף עשיר ומגוון של פונקציות פרימיטיביות הפועלות על מערכים כארגומנטים.

המערך ב-APL הנו מערך רבועי (rectangular array) ומכיל אוסף הומוגני ומסודר של אברים. אברי המערך משתייכים לאחד משני סוגי הנתונים המוגדרים בשפה א. מספרים ב. תוים, כל אבריו של מערך נתון משתייכים לאותו סוג נתונים (הומוגניות). אוסף האברים המרכיבים מערך הנו מסודר באופן שנתן לזהות כל איבר בודד ע"י סדרה של אינדקסים, דרגת המערך הנה מספר האינדקסים הנדרשים בכדי לאפיין איבר בודד של המערך. נתן לחשוב על דרגת המערך כבעל מספר ממדיו, לדוגמה וקטור המספרים 1 2 3 4 5 הנו מערך בעל מימד 1; טבלה של מספרים הנה מערך דו-מימדי, נדרשים שני אינדקסים בכדי לאפיין כל איבר בטבלה, האינדקס האחד מציין את מספר השורה בטבלה בעוד האינדקס השני מציין את מספר העמודה (למערך דו-ממדי נקרא גם מטריצה). סקלר הנו מקרה פרטי של מערך, הסקלר הנו מערך בעל דרגה 0, מכיל איבר בודד ולא נדרש אף אינדקס בכדי לזהותו במפורש. מאפיין נוסף של המערך הנו וקטור הצורה, וקטור זה הנו סדרה של מספרים, אורך הסדרה כדרגת המערך וכל איבר בוקטור זה מציין את מספר האברים במימד המתאים של המערך. לדוגמה וקטור הצורה של מערך חד ממדי (כלומר של וקטור) הנו מספר יחיד המציין את מספר האברים במערך; וקטור הצורה של מטריצה מכיל שני אברים המציינים את מספר השורות ומספר העמודות של המטריצה בהתאמה.

פונקציות סקלריות

השמוש במונח פונקציה ב-APL מקורו במלה אשר משמעותה ביצוע פעולה. פונקציה פרימיטיבית של השפה מבצעת פעולה מוגדרת על הארגומנט (ים) שלה ומייצרת תוצאה אשר יכולה לשמש כארגומנט לפונקציה פרימיטיבית אחרת. כל פונקציה פרימיטיבית של השפה מאופיינת על פי מספר הארגומנטים שלה; פונקציות מונדיות פועלות על ארגומנט יחיד, פונקציות אשר פועלות על שני ארגומנטים נקראות דיאדיות.
כבכל שפת תכנות אחרת קיימים ב- APL משתנים, המשתנים הנם שמות אשר נתן לשייך להם מערכים באמצעות משפטי הצבה, לדוגמה 5 4 3 2 1 ← A.

נבחן עתה את המשמעות של הבטוי $A+B$, בו A ו-B הנם שמות של משתנים והסימן + מייצג את הפונקציה הפרימיטיבית "חבור". כאשר A ו- B מייצגים ערכים סקלריים תהיה תוצאת הבטוי סכום שני המספרים כפי שמקובל בשפות תכנות אחרות. אולם כאשר A ו-B מייצגים מערכים בעלי דרגה גבוהה מ-0 מתפרש הבטוי כסכום האברים המתאימים של הארגומנטים, לדוגמה:

$A \leftarrow 1\ 2\ 3\ 4\ 5$

$B \leftarrow 2\ 3\ 1\ 5\ 4$

$A + B \Rightarrow 3\ 5\ 4\ 9\ 9$

דוגמה פשוטה זו מרמזת על מדת התמציתיות של הכתיבה ב- APL. בכדי להפיק את תוצאת החבור שבדוגמה נדרש בצוע חוג (loop) אשר אינו מופיע באופן מפורש בביטוי, משמעותה של עובדה זו היא כי נטל הטפול בפרטים "מיותרים" (דוגמת מנית מספר הפעמים שהחוג בוצע עד כה, ובדיקה לסיום החוג) מועבר מן המתכנת אל התכנית המפענחת. נציין גם כי התוצאה של הפונקציה הסקלרית הנה מערך אשר וקטור הצורה שלו זהה לזה של הארגומנטים (להוציא מקרים בהם אחד הארגומנטים הנו סקלר ואז וקטור הצורה שוה לזה של הארגומנט השני). הפונקציה הפרימיטיבית + הנה אחת מאוסף של פונקציות פרימיטיביות הנקראות פונקציות סקלריות (פונקציות אלו נקראות סקלריות מאחר ותוצאתן מוגדרת עבור ארגומנט (ים) סקלריים ומורחבות, כפי שראינו בדוגמה למעלה, למערכים). אוסף הפונקציות הסקלריות מכיל את הפונקציות האריתמטיות(+, -, ×, ÷), פונקציות לחשוב חזקות, לוגריתמים (בבסיס כל שהוא), פונקציות למציאת מינימום ומקסימום, פונקציות טריגונומטריות והיפרבוליות, פונקציות בולאניות ועוד. תת-קבוצה כללית וחשובה של הפונקציות הסקלריות הן פונקציות היחס. ב- APL סימני היחס ≠ > ≥ = ≤ < משמשים כפונקציות המחזירות את הערך 1 כאשר היחס מתקיים ו-0 במקרה ההפוך, ובכך נתנת האפשרות לכלול פונקציות אלו בתוך ביטויים אריתמטיים. השלוב שבין פונקציות היחס והפונקציות הבולאניות מאפשר כתיבתם של ביטויים מאד כלליים לחשובים מותנים וכתוצאה מכך פשוט רב במבנה התכניות ע"י מניעתם של משפטי GOTO במקרים רבים.

אופרטורים

בדוגמה למעלה ראינו צורה אחת של הרחבת תחום פעולתן של הפונקציות הסקלריות למערכים (ע"י בצוע הפעולה איבר אחר איבר). ב-APL קיימים אמצעים נוספים המגדילים באופן ניכר את יכולת הנצול של הפונקציות הסקלריות (דיאדיות) לגבי מערכים. אמצעים אלו נקראים אופרטורים ומאפשרים את בצוע הפעולה בין אברי הארגומנטים באופן שיטתי ובסדר שונה מזה שראינו למעלה.
נדגים את פעולתו של אופרטור ה- reduction המיוצג ע"י הסימן /. נניח כי הבעיה העומדת בפנינו היא לסכם את כל אבריו של וקטור המספרים A. בכדי להשיג אפקט זה מספיק לכתוב את הביטוי +/A. לבטוי זה שני מרכיבים, האחד הנו הארגומנט A, והשני הנו הפונקציה המבצעת את הפעולה הרצויה "סכום כל אברי A", את הפונקציה הנדרשת אנו מגדירים באמצעותו של האופרטור / והפונקציה הסקלרית הפרימיטיבית +. האופרטור / מרחיב את תחום הפעולה של הפונקציה הסקלרית אשר לשמאלו כאילו וכתבנו את סימן הפונקציה הסקלרית בין כל שני אברים של הארגומנט לאורך המימד האחרון שלו, התוצאה המתקבלת הנה מערך שדרגתו נמוכה ב- 1 מדרגת הארגומנט. באופן דומה נתן להגדיר את הפונקציה ⌊/ (הסימן ⌊ הינו הפונקציה הסקלרית למציאת המינימום של הארגומנטים) אשר תחזיר כתוצאה את האיבר המינימלי כאשר נפעיל אותה על וקטור מספרים.

עתה נניח כי T היא טבלה של מחירים למוצרים, שורות הטבלה מתאימות למוצרים ואלו עמדותיה מתאימות לספקים, כך שהאיבר במקום ה I,J מציין את המחיר שדורש ספק J עבור המוצר I. ברצוננו לרכוש את כל המוצרים המופיעים בטבלה במחיר כולל נמוך ביותר (כלומר לרכוש כל מוצר מהספק הדורש מחיר נמוך ביותר עבור אותו מוצר).
הביטוי +/⌊/T מחשב מחיר זה. בביטוי זה אנו רואים שתי פעולות reduction, סדר הבצוע של הפעולות ב- APL הנו מימין לשמאל ולכן ראשית מבצע המפענח את הפונקציה ⌊/T, התוצאה של ביטוי זה הנה וקטור של המחירים הנמוכים ביותר עבור כל מוצר, הביטוי +/ פועל על תוצאת ביניים זו ונותן את התשובה הנדרשת.
בשפה מוגדרים עוד שלשה אופרטורים, inner-product אשר סימנו . , outer-product אשר סימנו ∘. ו- scan אשר סימנו \. אופרטורים אלו בדומה ל- reduction מגדירים כל אחד מרשם כללי להרחבת תחום פעולתן של הפונקציות הסקלריות ומאפשרים לבטא באופן תמציתי ואחיד קבוצה גדולה מאד של פונקציות חדשות.

פונקציות מעורבות

הפונקציות האחרות של APL נקראות mixed functions: פונקציות אלו מוגדרות לגבי ארגומנטים שהנם מערכים, באמצעות הפונקציות המעורבות נתן לייצר ולארגן מערכים בהתאם לצורה נדרשת, לבחור קבוצות אברים מתוך מערכים נתונים, לייצר מערכים המשמשים בעקר את פונקציות הבחירה, להפוך נתוני תוים למספרים, להפוך נתונים מספריים לתוים ולערוך אותם כרצוי ועוד. שלא כמו במקרה של הפונקציות הסקלריות אין כלל אחיד להתנהגותן של הפונקציות המעורבות, לכל אחת מהפונקציות הגדרה

ספציפית לתחום הארגומנטים שלה ולצורת המערך המתקבל כתוצאה מהפעלתה.

לדוגמה ι הנו סימן המייצג שתיים מן הפונקציות המעורבות, (ב- APL משמשים סימנים רבים לייצוג שתי פונקציות בו זמנית, למשל הסימן - מייצג פונקציה דיאדית שמשמעותה חסור, ופונקציה מונדית שמשמעותה הפוך הסימן), בהופעתו המונדית משמש סימן זה כפונקציה היוצרת סדרת מספרים שלמים, לדוגמה 5 4 3 2 1 <= ι5; ובהופעתו הדיאדית פונקציה הנקראת index of ומחזירה את האינדקס של אברי הארגומנט הימני בתוך הארגומנט השמאלי לדוגמה

1 5 7 3 ι7 5 => 3 2

הסימן , מייצג גם הוא שתי פונקציות מעורבות, נתייחס לפונקציה הדיאדית , המייצגת את פעולת השרשור (catenation); פונקציה זו משרשרת את אברי שני הארגומנטים שלה לדוגמה

1 2 3 4 ,5 6 7 => 1 2 3 4 5 6 7

הנה שרשור של שני וקטורים, בעוד

```
1 2 3             1 2 3 5
4 5 6 , 5 7 9 =>  4 5 6 7
7 8 9             7 8 9 9
```

הנה דוגמה של שרשור וקטור למטריצה לאורך המימד האחרון של המטריצה (המימד האחרון הנו ברירת המחדל לצורך השרשור), את השרשור נתן לבצע גם לאורך מימד אחר המוגדר באופן מפורש, הדוגמה הבאה מראה כיצד נתן לשרשר מטריצה ווקטור לאורך המימד הראשון של המטריצה

```
1 2 3                    1 2 3
4 5 6 ,[1] 5 7 9 =>      4 5 6
7 8 9                    7 8 9
                         5 7 9
```

הפונקציה המונדית ι מוגדרת רק לארגומנט שהנו סקלר או וקטור בעל איבר אחד ותוצאתה תמיד וקטור. הפונקציה הדיאדית ι מוגדרת עבור ארגומנט שמאלי שהנו וקטור וארגומנט ימני שהנו מערך כל שהוא, צורת התוצאה כצורת הארגומנט הימני. הפונקציה הדיאדית , מוגדרת לארגומנטים כל שהם בתנאי שוקטורי הצורה שלהם מקיימים דרישות מסוימות, צורת התוצאה תלויה במימד שלאורכו מתבצעת הפעולה.

<u>תכונות כלליות נוספות</u>

כפי שראינו בסעיפים הקודמים מאפשרות הפונקציות הפרימיטיביות של APL כתיבה מאד תמציתית של תכניות. הגדרתן של הפונקציות הנה כללית למדי, וכל פונקציה כזו מייצגת למעשה מגוון של פעולות אפשריות כתת מקרים בהתאם לתכונות הארגומנטים (במיוחד צורתם). פורמלית מבחינה השפה רק בשני סוגי הנתונים מספרים ותוים, אולם במימוש השפה מקובל לייצג את המספרים באחת משלוש צורות הצגה פנימית, שלמים, ממשיים ובולאניים. התכנית המפענחת הופכת את הנתונים המספריים מצורת הצגה אחת לאחרת בהתאם לנדרש, לדוגמה אם בבטוי A+B אחד הארגומנטים מיוצג כשלם והשני כממשי

אופרטורים

בדוגמה למעלה ראינו צורה אחת של הרחבת תחום פעולתן של הפונקציות הסקלריות למערכים (ע"י בצוע הפעולה איבר אחר איבר). ב-APL קיימים אמצעים נוספים המגדילים באופן נכר את יכולת הנצול של הפונקציות הסקלריות (דיאדיות) לגבי מערכים. אמצעים אלו נקראים אופרטורים ומאפשרים את בצוע הפעולה בין אברי הארגומנטים באופן שיטתי ובסדר שונה מזה שראינו למעלה.

נדגים את פעולתו של אופרטור ה- reduction המיוצג ע"י הסימן /. נניח כי הבעיה העומדת בפנינו היא לסכם את כל אבריו של וקטור המספרים A. בכדי להשיג אפקט זה מספיק לכתוב את הבטוי A/+. לבטוי זה שני מרכיבים, האחד הנו הארגומנט A, והשני הנו הפונקציה המבצעת את הפעולה הרצויה "סכום כל אברי A", את הפונקציה הנדרשת אנו מגדירים באמצעותו של האופרטור / והפונקציה הסקלרית הפרימיטיבית +. האופרטור / מרחיב את תחום הפעולה של הפונקציה הסקלרית אשר לשמאלו כאילו וכתבנו את סימן הפונקציה הסקלרית בין כל שני אברים של הארגומנט לאורך הממד האחרון שלו, התוצאה המתקבלת הנה מערך שדרגתו נמוכה ב- 1 מדרגת הארגומנט. באופן דומה נתן להגדיר את הפונקציה ⌊/ (הסימן ⌊ הינו הפונקציה הסקלרית למציאת המינימום של הארגומנטים) אשר תחזיר כתוצאה את האיבר המינימלי כאשר נפעיל אותה על וקטור מספרים.

עתה נניח כי T היא טבלה של מחירים למוצרים, שורות הטבלה מתאימות למוצרים ואלו עמדותיה מתאימות לספקים, כך שהאיבר במקום ה I,J מציין את המחיר שדורש ספק J עבור המוצר I. ברצוננו לרכוש את כל המוצרים המופיעים בטבלה במחיר כולל נמוך ביותר (כלומר לרכוש כל מוצר מהספק הדורש מחיר נמוך ביותר עבור אותו מוצר). הביטוי T⌊/+/ מחשב מחיר זה. בביטוי זה אנו רואים שתי פעולות reduction, סדר הבצוע של הפעולות ב- APL הנו מימין לשמאל ולכן ראשית מבצע המפענח את הפונקציה T⌊/, התוצאה של ביטוי זה הנה וקטור של המחירים הנמוכים ביותר עבור כל מוצר, הביטוי +/ פועל על תוצאת ביניים זו ונותן את התשובה הנדרשת.

בשפה מוגדרים עוד שלשה אופרטורים, inner-product אשר סימנו . , outer-product אשר סימנו .∘ ו- scan אשר סימנו \. אופרטורים אלו בדומה ל-reduction מגדירים כל אחד מרשם כללי להרחבת תחום פעולתן של הפונקציות הסקלריות ומאפשרים לבטא באופן תמציתי ואחיד קבוצה גדולה מאד של פונקציות חדשות.

פונקציות מעורבות

הפונקציות האחרות של APL נקראות mixed functions: פונקציות אלו מוגדרות לגבי ארגומנטים שהנם מערכים, באמצעות הפונקציות המעורבות נתן לייצר ולארגן מערכים בהתאם לצורה נדרשת, לבחור קבוצות אברים מתוך מערכים נתונים, לייצר מערכים המשמשים בעקר את פונקציות הבחירה, להפוך נתוני תוים למספרים, להפוך נתונים מספריים לתוים ולערוך אותם כרצוי ועוד. שלא כמו במקרה של הפונקציות הסקלריות אין כלל אחיד להתנהגותן של הפונקציות המעורבות, לכל אחת מהפונקציות הגדרה

ספציפית לתחום הארגומנטים שלה ולצורת המערך המתקבל כתוצאה מהפעלתה.

לדוגמה ⍳ הנו סימן המייצג שתיים מן הפונקציות המעורבות, (ב- APL משמשים סימנים רבים לייצוג שתי פונקציות בו זמנית, למשל הסימן - מייצג פונקציה דיאדית שמשמעותה חסור, ופונקציה מונדית שמשמעותה הפוך הסימן), בהופעתו המונדית משמש סימן זה כפונקציה היוצרת סדרת מספרים שלמים, לדוגמה 5 4 3 2 1 <= ⍳5, ובהופעתו הדיאדית פונקציה הנקראת index of ומחזירה את האינדקס של אברי הארגומנט הימני בתוך הארגומנט השמאלי לדוגמה

1 5 7 3 ⍳7 5 => 3 2

הסימן , מייצג גם הוא שתי פונקציות מעורבות, נתייחס לפונקציה הדיאדית , המייצגת את פעולת השרשור (catenation), פונקציה זו משרשרת את אברי שני הארגומנטים שלה לדוגמה 1 2 3 4 ,5 6 7 => 1 2 3 4 5 6 7

הנה שרשור של שני וקטורים, בעוד

```
1 2 3                1 2 3 5
4 5 6 , 5 7 9 => 4 5 6 7
7 8 9                7 8 9 9
```

הנה דוגמה של שרשור וקטור למטריצה לאורך המימד האחרון של המטריצה (המימד האחרון הנו ברירת המחדל לצורך השרשור), את השרשור נתן לבצע גם לאורך מימד אחר המוגדר באופן מפורש, הדוגמה הבאה מראה כיצד נתן לשרשר מטריצה ווקטור לאורך המימד הראשון של המטריצה

```
1 2 3                      1 2 3
4 5 6 ,[1] 5 7 9 => 4 5 6
7 8 9                      7 8 9
                           5 7 9
```

הפונקציה המונדית ⍳ מוגדרת רק לארגומנט שהנו סקלר או וקטור בעל איבר אחד ותוצאתה תמיד וקטור. הפונקציה הדיאדית ⍳ מוגדרת עבור ארגומנט שמאלי שהנו וקטור וארגומנט ימני שהנו מערך כל שהוא, צורת התוצאה כצורת הארגומנט הימני. הפונקציה הדיאדית , מוגדרת לארגומנטים כל שהם בתנאי שוקטורי הצורה שלהם מקיימים דרישות מסוימות, צורת התוצאה תלויה במימד שלאורכו מתבצעת הפעולה.

<u>תכונות כלליות נוספות</u>

כפי שראינו בסעיפים הקודמים מאפשרות הפונקציות הפרימיטיביות של APL כתיבה מאד תמציתית של תכניות. הגדרתן של הפונקציות הנה כללית למדי, וכל פונקציה כזו מייצגת למעשה מגוון של פעולות אפשריות כתת מקרים בהתאם לתכונות הארגומנטים (במיוחד צורתם). פורמלית מבחינת השפה רק בשני סוגי הנתונים מספרים ותוים, אולם במימוש השפה מקובל לייצג את המספרים באחת משלוש צורות הצגה פנימית, שלמים, ממשיים ובולאניים. התכנית המפענחת הופכת את הנתונים המספריים מצורת הצגה אחת לאחרת בהתאם לנדרש, לדוגמה אם בבטוי $A+B$ אחד הארגומנטים מיוצג כשלם והשני כממשי

יהורגם הארגומנט שהצגתו בשלמים להצגה ממשית והפעולה תתבצע ביחידה האריתמטית המתאימה של המחשב. הפוך זה אינו דורש את התערבות המתכנת ומבוצע באופן אוטומטי. חלק מהפונקציות הפרימיטיביות אדיש לסוג הארגומנטים שלהן (למשל פונקצית השרשור מוגדרת לכל הסוגים) בעוד פונקציות אחרות דורשות שהארגומנטים יהיו מסוג מסוים, לדוגמה פעולת החבור תתבצע רק עבור ארגומנטים שהם מסוג "מספר". התכנית המפענחת בודקת בכל מקרה את הדרישות לגבי הארגומנטים הן מבחינת הסוג והן מבחינת הצורה בהתאם להגדרת השפה (לא נתן למשל לחבר מטריצה עם מערך תלת ממדי), כאשר המפענח מאתר חריגה מהדרישות מופסק בצוע התכנית ונשלחת הודעת שגיאה למשתמש, תוך כדי ציון מדויק של סוג השגיאה ומיקומה המדויק בתוך הבטוי בו נתגלתה. בהיות המערכת אינטראקטיבית נתנת למשתמש האפשרות לבדוק את הגורמים לשגיאה, לתקנם ולהורות למערכת להמשיך את הבצוע ממקום רצוי.

תכונה חשובה נוספת של השפה מתבטאת בחוסר משפטי הצהרה (declaration statements). נתן לשייך מערך כל שהוא לכל משתנה המופיע בתכנית, יתר על כן למשתנה מסוים נתן לשייך במהלך הבצוע מערכים בעלי צורה וסוג כל שהוא בלי כל קשר לתכונות הערך הנכחי של המשתנה. לכן לדוגמה משפט ההצבה $A \leftarrow A,B$, בו A ו-B הנם וקטורים בעלי אורך 5 ו- 3 בהתאמה, יגדיר את A בוקטור בעל 8 אברים.

נהול הזכרון הנדרש לאכסון תוצאות ביניים וערכי משתנים מבוצע אוטומטית ע"י התכנית המפענחת. בדוגמה הקודמת $A \leftarrow A,B$ יוקצה זכרון עבור וקטור התוצאה, ויוחזר הזכרון שהיה תפוס ע"י הערך של A, לפני בצוע משפט ההצבה, למאגר הזכרון הפנוי.

<u>יתרונות וחסרונות עקריים</u>

יחודה העקרי שלAPL מתבטא בעצמה הטמונה במדת תמציתיותה. העושר הרב בפונקציות פרימיטיביות לבצוע פעולות רבות ומגוונות על מערכים, כלליות הגדרתן של הפונקציות הפרימיטיביות ויחד עם זאת שמירה על מספר מצומצם של כללים וחוקים הנם הבסיס לכתיבה מאד תמציתית ומדוקת. רבים טוענים כי מדת תמציתיות זו הנה בעכריה של השפה מאחר ולא נתן לקרוא ולהבין את ביטוייה. לדעה זו אין אנו מסכימים, אמנם במבט ראשון נראים ביטויי APL כבלתי נתנים להבנה אולם תוך כדי רכישת נסיון עם השפה ולאחר שמתרגלים לאוסף הסימנים המיוחד לשפה מסתבר כי נסוח הביטויים משקף בדיוק את ה"מהי" ולא את ה"איך". לדוגמה הביטוי $\lceil/V$ פירושו מצא את האיבר המכסימלי של הוקטור V (כאשר V הנו וקטור) ואינו דורש כתיבה של נוהל לבצוע מטרה זו. אמנם נכון הוא כי עצמתן של הפונקציות הפרימיטיביות מאפשרת כתיבה של ביטויים מאד מסובכים אשר מדת בהירותם מוטלת בספק, אולם גם בשפות תכנות נמוכות מ-APL נתן לכתוב תכניות אשר המבנה הלוגי שלהן ואופן כתיבתן אינו מאפשר מעקב והבנה, מדת הבהירות של התכנית אינה תכונה העבועה בשפה אלא באופן בו משתמשים בשפה.

Streeter [9]בצע ניסוי ובו מדד את משך הזמן הנדרש לכתיבת תכניות במספר שפות תכנות. מדידות אלו מראות כי זמן הכתיבה וניפוי השגיאות ב-APL הנו ביחס של 3:1 למשך הזמן הנדרש לכתיבת אותן תכניות ב- FORTRAN. האורך הממוצע של תכנית הכתובה ב- APL קטן באופן משמעותי מתכנית שקולה הכתובה ב- FORTRAN.
המדידות של Saal and Weiss [10] מראות כי היחס בין האורך הממוצע של תכנית FORTRAN , שהנו כ-100 כרטיסים, לאורך הממוצע של תכנית APL הנו 8:1.

יחד עם זאת אין השפה APL אמצעי שכולו יתרונות ומעלות. אחת המגבלות של APL נובעת מן העובדה כי מבנה הנתונים היחיד של השפה הנו מערך רבועי והומוגני. כאשר דורש יישום מסוים מבני נתונים אשר אינם נתנים להצגה באופן טבעי כמערכים הומוגניים ומרובעים, נאלץ המתכנת ב- APL לכפות את המערך על מבני הנתונים הטבעיים לבעיה ולכתוב שגרות המרחיבות למעשה את APL לטפול במבני הנתונים הרצויים. נושא מבנה הנתונים זכה לתשומת לב רבה בקרב משתמשי APL ולהצעות שונות להרחבת השפה ולהכללה של מבני נתונים נוספים; ראה לדוגמה [11].

הנקודה הכאובה ביותר בקשר למערכות APL הנה אי היעילות בזמני הבצוע של תכניות APL. כפי שראינו מבוססת מערכת APL על תכנית מפענחת ולא על מהדר (Compiler), עובדה זו הנה הגורם הישיר לזמני הבצוע הארוכים של תכניות APL. באשר ליעילות הבצוע נעיר שתי הערות, ראשית ככל שתגדל ההשקעה בפתוח מערכות APL נוספות נתן לצפות כי מערכות אלו תהיינה יעילות יותר באופן משמעותי, לדוגמה כבר כיום קיימת גרסה של APL הכתובה במקרו-תכנות ומתאימה למחשבי IBM/370 מדגם 145, ראה [12], גרסה זו יעילה פי 2 עד 20 מן הגרסה הכתובה באסמבלר 370. שנית, בתחומים שונים משך פתוח התכנה, גמישות התכניות ונצול אינטראקטיבי של התכנה הנם הגורמים המכריעים בבחירת השפה המתאימה, בעוד זמני הבצוע הנם בעלי חשיבות משנית. הירחון EDP Analyses [13] יהקדיש את אחת מהוצאותיו האחרונות לסקירה של השמוש ב- APL בתחום מערכות העזר הניהוליות. במערכות אלו נדרשת בד"כ מהירות רבה בפתוח התכנה,APL נבחרה ככלי המתאים למילוי הדרישה על אף יעילותה הנמוכה.

<u>רשימת מקורות</u>

1. K. E. Iverson, A Programming Language, Wiley, New York, 1962.
2. H. Hellerman, "Experimental Personalized Array Translator", Communication of the ACM 7,433 (July 1964).
3. A. D. Falkoff and K. E. Iverson, "The APL\360 Terminal System", Symposium on Interactive Systems for Experimental Applied Mathematics, eds. M. Klerer and J. Reinfelds, Academic Press, New York, 1968.
4. L.M. Breed and R. H. Lathwell, "The implementation of APL\360", (same as 3).

5. A. D. Falkoff and K. E. Iverson, APL\360 User's Manual, IBM Corporation, (GH20-0683) 1968.

6. APL PLUS, File Subsystem Instruction Manual, Scientific Time Sharing Corporation, 7316 Wisconsin Avenue, Bethesda, Md. 20014

7. APLSV, User's Guide (SH20-1460) IBM Corporation.

8. APL\700, User's Reference Manual, Document No. 5000813, Burroughs Corporation.

9. D. N. Streeter, "Cost Benefit Evaluation of Scientific Computing Services", IBM Systems Journal, Vol 11, No. 3 pp.219-233.

10. H.J. Saal and Z. Weiss, "Some Properties of APL Programs", APL 75 Congress Proceedings, pp 292-297.

11. J. E. Mezei, "Uses of General Arrays and Operators", Proceedings of the Sixth International APL User's Conference, pp 334-348.

12. A. Hassit and L.E. Lyon, "Direct execution of APL on IBM\360", APL 75 Congress Proceedings, pp 188-191.

13. "APL and Decision Support Systems", EDP Analyzer, Vol 14, No 5. May 1976.

ספרי למוד ל- APL

14. R. P. Polivka and S. Pakin, APL: The Language and Its Usage, Prentice Hall, 1975.

15. L. Gilman and A. J. Rore, APL An Interactive Approach, John Wiley and Sons 1974.

16. H. Katzan, APL Programming and Computer Techniques, Van Norstrand Reinhold, 1970.

17. L. D. Gray, A Course in APL\360 with Applications, Addison Wesly, 1973.

נהול מרפאת חוץ של בי"ח ע"י מערכת A P L

ד"ר שלמה בכר, ענה מרחב, פרופ' הנרי נויפלד
מכון הלב, המרכז-הרפואי ע"ש חיים שיבא.

ד"ר מאיר שניידר, דניאל ליברמן
אוניברסיטת בר-אילן

א. מבוא

במרפאת החוץ של מכון הלב במרכז הרפואי ע"ש חיים שיבא, מבקרים כ-300 חולים בשבוע. בנסיון הנוכחי נלקחו רק 50 חולים בשבוע. נהול התיקים והמעקב אחר מספר כה רב של אנשים מהווים בעיה אדמיניסטרטיבית ורפואית רצינית למדי. כאשר מגיע הנבדק למרפאה, חייב תיקו להיות אצל הרופא. כשמסתיימת הבדיקה, חייבות הערות הרופא להרשם בתיק והתיק צריך לחזור לתיקיה. הנבדק מקבל הזמנה לבקור הבא וכן לכל הבדיקות הנוספות אותן ביקש הרופא. תוצאות כל הבדיקות חייבות להרשם בתיק עד לבקורו הבא של הנבדק אצל הרופא.

אם מובא החולה באופן דחוף לבית החולים, צריכים לאתר את תיקו כדי שהרופא המטפל יוכל לעיין בהיסטוריה הרפואית של החולה.

אם מאושפז החולה במחלקה אחרת, על מכון הלב להעביר לאותה מחלקה מידע מתיקו של החולה.

נוסף לבעיה האדמיניסטרטיבית, קיימת בעיה מדעית רפואית. בכל התיקים האלה נמצא מאגר עשיר מאד של מידע על הממצאים הרפואיים והתגובות לטפולים שונים בנסיבות שונות. בכדי לנתח נתונים סטטיסטיים

רבי ערך אלה, יש צורך לשלוף את הנתונים מכל תיק ותיק.

בכדי להקל על הבעיות האדמיניסטרטיבית והבעיות הרפואיות באופן היעיל ביותר, הוחלט לנסות להקים מערכת ממוחשבת לנהול המרפאה. מאמר זה בא לדווח על אותו ניסוי.

המערכת נכתבה ב-APLSV בשל הסיבות הבאות:

1. המערכת צריכה להיות אינטראקטיבית, לפחות בשלב העידכון, כי רק אנשי המרפאה רשאים לתקן שגיאות ולעדכן נתונים והם נמצאים בבית החולים ולא במרכז-המחשבים של האוניברסיטה. בגלל מספרם הגדול של הנבדקים הבאים כל יום, אי אפשר לחכות להדפסות BATCH כדי לקבל דו"ח שבועיים.

2. המערכת צריכה לעבוד עם קבצים. בגלל מספרם הגדול של הנבדקים והפריטים, אי אפשר לשמור את המידע בתוך שטחי העבודה של APL.

3. כיון שתקציב התכנות היה מצומצם, צריכה היתה המערכת להכתב בזמן קצר. בשפת APL, כל המערכת (כ-600 שורות בכ-50 פונקציות) נכתבה ונופתחה תוך שלושה חדשים ע"י מתכנת אחד.

תאור הקבצים

ישנם 5 קבצים במערכת. כל קובץ מורכב ממטריצות (בלוקים), וכל מטריצה מורכבת משורות (רשומות) באורך קבוע. עבור כל קובץ, ישנו אלגוריתם המחשב את מספר המטריצה. נתון תאריך הביקור או מספר הזהוי של הנבדק.

הקבצים הם:

1. דמוגרפי

כל שורה מכילה מידע אלפה-נומרי על נבדק אחד: מספר זהוי, שם, כתובת, גיל, מין, הרופא או קופ"ח המפנה, ועוד. כל מטריצה מכילה את השורות של בעלי מספרי זהוי מסוימים, כך שהאלגוריתם מחשב ממספר הזהוי את מספר המטריצה המכילה את השורה של אותו מספר זהוי.

2. בקורים אצל רופא

כל שורה מכילה מידע נומרי על נבדק אחד: מספר זהוי, קוד הרופא אליו הוא מוזמן, השעה וקודים מסוימים. כל מטריצה מכילה את כל ההזמנות לתאריך אחד, כך שהאלגוריתם מחשב מתאריך הביקור את מספר המטריצה המכילה את ה"יומן" של אותו תאריך, וכך, אין צורך לאכסן את התאריך עצמו בשורה של הנבדק.

3. תורים לבדיקות

המבנה זהה למבנה של הקובץ הקודם, אבל במקום קוד הרופא מופיע קוד הבדיקה עבורה מוזמן הנבדק.

4. תוצאות הבקור אצל הרופא

כל שורה מכילה מידע נומרי על נבדק אחד: מספר זהוי, תאריך הבקור, קוד הרופא, קודים של אבחנות וטפולים ועוד. השורות מחולקות למטריצות

- עפ"י מספר הזהוי במו בקובץ הדמוגרפי.

5. <u>תוצאות של בדיקות</u>

המבנה זהה למבנה של הקובץ הקודם, אבל במקום קוד הרופא וקודים של אבחנות וטפולים מופיעים קוד הבדיקה וקודים של תוצאות הבדיקה.

ג. <u>תאור הפונקציות</u>

הפונקציות הראשיות מתחלקות לשלשה סוגים - קלט, עדכון, ופלט. אפשר לבצע כל אחד משלשת הסוגים על כל אחד משלשת הסוגים של קבצים - דמוגרפי, בקורים (רופא,בדיקה), ותוצאות (רופא, בדיקה). פונקצית הקלט לתוצאות בודקת אם הקודים של התוצאות מוכרים לה. אם לא, צריכה הפקידה להגדיר את הקודים החדשים במילון. כמו-כן, בודקת הפונקציה אם התוצאות נדבעות מבקור שהיה רשום בקובץ הבקורים. אם לא, קוראת הפונקציה לפונקצית הקלט לבקורים בכדי לרשום את הבקור בקובץ הבקורים. אחר כך, רושמת הפונקציה את התוצאה בשורה של הנבדק בקובץ התוצאות.

פונקצית הקלט לבקורים בודקת אם התאריך הוא אחד התאריכים בהם המרפאה או המעבדה פתוחות, ואם קוד הרופא או הבדיקה מוכר לה. אם לא, יש להגדיר את הקוד במילון. הפונקציה בודקת גם אם הנבדק נמצא בקובץ הדמוגרפי. אם לא, קוראת הפונקציה לפונקצית הקלט לדמוגרפי. אחר כך, רושמת הפונקציה את פרטי הבקור בקובץ הבקורים בתאריך המתאים.

פונקצית הקלט לדמוגרפי מדפיסה טופס במסוף והפקידה מתקתקת את המידע הדרוש. קיימת אפשרות לדלג על חלק מהנתונים אבל נתונים אחרים הם הכרחיים. הפונקציה בודקת שביאות לוגיות ורושמת את המידע בקובץ

הדמוגרפי לפי מספר תעודת הזהות.

פונקציית העדכון מאפשרות שנוי של כל שדה בכל שורה של כל קובץ. נבדקת, כמובן, אפשרות של שגיאות לוגיות בפרטים החדשים.

פונקציית הפלט מאפשרות שליפה מכל קובץ לפי קריטריונים מסוימים וגם הדפסת מידע מתוך כמה קבצים. למשל, אפשר לקבל רשימה של כל האנשים שהוזמנו לרופא מסוים בתאריך מסוים או לכל הרופאים בתאריך מסוים. אפשר לקבל רשימה של כל ההזמנות של נבדק אחד. אפשר לקבל רשימה של כל התוצאות של נבדק אחד, וכן, אם המכתב כבר נשלח לרופא המפנה. אפשר לקבל רשימה של כל הנבדקים אשר מכתביהם טרם נשלחו, וכן הלאה.

נוסף לפונקציות המנהליות הללו, ישנן פונקציות סטטיסטיות. אלו הן פונקציות פלט המאפשרות לוח או ספירה של כל הנבדקים שקבלו טפול מסוים או שתוצאותיהם היו זהות, או-מבין אלה שקבלו אותו טפול - התפלגות התוצאות, וכן הלאה.

כל הפונקציות קוראות לפונקציות קלט/פלט שנכתבו במיוחד למערכת הזאת, ואשר משתמשות באלגוריתמים של הקבצים האלה. כל הפונקציות הללו קוראות לפונקציות קלט/פלט כלליות ובסיסיות יותר אשר מחקות את פונקציות הקלט/פלט של APL PLUS.

ד. הטפול בקבצים

הקבצים נבנו ע"י הפקודה:

CTL←'SW ',PARAM

כאשר PARAM מכיל את כל הפרמטרים ב-JCL לרבות *SPACE*

וכאשר CTL הנו משתנה משותף למערכת ההפעלה.

הפקודה:

J←CTL

מכניסה קוד שגיאה ל-J (כאשר הקוד 0 מסמן "הכל בסדר").

לאחר הבניה, מכניסים לקובץ מטריצות ריקות השורות במספרן למספר המכסימלי של מטריצות שיכנסו לקובץ. כתיבת המטריצות הריקות הללו מאפשרת הוספת שורות (רשומות) חדשות מבלי להגדיר מטריצות (בלוקים) חדשות. תכנית הכתיבה קוראת את המטריצה המתאימה, מוסיפה את השורה החדשה בסוף המטריצה, וכותבת את המטריצה שוב במקומה.

בתחילת כל עבודה ליד המסוף, נפתחים הקבצים ע"י הפונקציה START אשר קוראת לפונקציה FTIE עבור כל קובץ וקובץ ונותנת לכל אחד מספר משלו. למשל:

```
        ∇START
[1]     'A12345.DEMOGRAF' FTIE 1
[2]     →(J≠0)/ERROR
[3]     'A12345.BIKUROFE' FTIE 2
[4]     →(J≠0)/ERROR
```

הפונקציה FTIE מגדירה שני משתנים משותפים המתיי אל מספר הקובץ:

```
    ∇DSN FTIE NUMBER;I;FN
[1]   I←370 ⎕SVO 'DAT',FN←⍕NUMBER
[2]   I←370 ⎕SVO 'CTL',FN
[3]   I←(4⍴1) ⎕SVC 'CTL',FN
[4]   I←(4⍴0) ⎕SVC 'DAT',FN
[5]   ⍎'CTL',FN,'←IRW UNIT=3330,VOL=SER=ABCDEF,DISP=OLD,
                                                DSN=',DSN
[6]→(0≠J←1↑,⍎'CTL',FN)/ERROR
[7]   →0
[8] ERROR:QLE[1+J;]∇
```

ובמשך כל זמן העבודה ליד המסוף, אפשר לקרוא ולכתוב בכל קובץ באמצעות

שני המשתנים האלה. למשל, הפונקציה WRITE היא פונקצית קלט/פלט

ספציפית למערכת זו:

```
     ∇R WRITE FILEBLOCK;FN;FILE;BLOCK;K
[1]   FN←⍕FILE←1↑FILEBLOCK
[2]   BLOCK←⍎'ALGORITHM',FN,' 1↓FILEBLOCK'
[3]   A←FREAD K←FILE,BLOCK
[4]   →(0≠J)/ERROR
[5]   (A,[1]R) FREPLACE K
```

כאשר ALGORITHM1 וכו' הן פונקציות המחשבות את מספר המטריצות, וכן,

FREAD ו-FREPLACE הן פונקציות כלליות המחקות את הפונקציות

ב- APLPLUS. למשל:

```
     ∇R FREPLACE FILEBLOCK;FN
[1]   ⍎'DAT',(FN←⍕1↑FILEBLOCK),'←R'
[2]   ⍎'CTL',FN,'←1 ',⍕1↓FILEBLOCK
[3]   →(0≠J←1↑,⍎'CTL',FN)/ERROR
[4]   →0
[5] ERROR:QLE[1+J;]∇
```

ה. חיפוש ושליפה ב- APLSV

המשימה העיקרית של כל המערכת המינהלית היא חיפוש ושליפה. בשפה APL מתבצעות הפעולות הללו ע"י פקודות פשוטות מאד: ι לחיפוש ו-/ לשליפה.

למשל, בכל הקבצים, העמודה הראשונה של המטריצה מכילה את מספר תעודת הזהות של הנבדק. בכדי להדפיס את השורה הראשונה המכילה מספר זהוי מסוים, מספיק לכתוב:

```
A[A[;1]⍳ID;]
```

בכדי להדפיס את כל השורות המכילות מספר זהוי מסוים, מספיק לכתוב:

```
(A[;1]=ID)⌿A
```

בכדי לקבל את כל השורות המקיימות תנאי מסוים (למשל, גיל מעל 40, מין זכר, וקוד אבחנה מסוים), מכניסים את התנאי למשתנה הגלובלי <u>TNAI</u> ומבצעים תכנית שבתוכה כתוב:

```
→(0=∨/I←⍎TNAI)/NONE
R←I⌿A
```

הודות לפקודות הפשוטות הללו, אפשר היה לכתוב את כל המערכת הזו ב-600 שורות בלבד ובזמן כה קצר.

ו. מסקנות

כאמור, נכתבה מערכת זו כניסוי. מסקנות הניסוי:

1. APLSV היא שפה יעילה לכתיבת והרצת מערכת זו.

2. במערכת זו, כתב וניפה את כל הפונקציות מתכנת אחד, ולאחר-מכן, הפסיק לעבוד בפרויקט. הפקידה יכולה אמנם להשתמש בפונקציות הללו, אך במשך הזמן הועלתה דרישה לפונקציות חדשות. הקבצים החלו להתמלא והיה צורך להעתיקם לסרט ולבנות קבצים חדשים במקומם. כלומר, צריך להיות מתכנת צמוד למערכת גם לאחר שלב הכתיבה.

3. הקלט והפלט אינם חייבים להתבצע ב- Real Time אלא העדכונים בלבד. יתכן שההוצאות הנגרמות כתוצאה מכתיבת כל המערכת ב- APL הנן גבוהות מדי, כאשר אפשר להריץ חלק ניכר ממנה ב- BATCH ורק העדכון צריך להיות אינטראקטיבי. כמו-כן, יתכן שלצורך עדכונים בלבד נתן להשתמש ב- TSO במקום APL.
השיקול שהנחה אותנו הוא: ריצות ב- BATCH הן אולי זולות יותר מריצות ב-APL , ואינן נפגעות ע"י נפילות המחשב. (הדבר נכון במיוחד אם רוצים לכלול את כל חולי המרפאה (300) במערכת המחשב ולא להסתפק רק במדגם קטן). אך, זמן התכנות והנפוי קצר יותר ב- APL מאשר בקובול או בשפת BATCH אחרת.
כך שלצורך הנסוי - התחלתו מוקדם ככל האפשר ללא תקציב גדול עבור מתכנת - היה זה שיקול נכון לכתוב ב-APL . כשיהיה מתכנת צמוד למערכת, יתכן שכדאי יהיה להעבירה ל- BATCH (פרט לעדכונים).

4. הסברה מתאימה איפשרה שילוב הרופאים בפרויקט ללא קשיים ושיתוף פעולתם היה משביע רצון בהחלט. הצורך בהעברת מסקנותיהם למחשב הביא לדיוק יתר בהגדרותיהם, הקפדה בניסוח וקיצור במשך הטפול של תיק החולה.

בנית תכנית שבה מעורבות הרופא הנה קצרה ביותר הוכיחה את עצמה מבחינה זו שהכנסת המחשב למרפאה לא נתקבלה כמטרד אלא כאמצעי עזר חשוב לעתיד.

צדדים אופייניים לפיתוח תוכנה בשפת APL

1. מבוא

המאמר מציג לקחי עבודה ב- APL שנלמדו תוך כדי שימוש בשפה זו לפיתוח מערכי תוכנה. מערכים אלה הם בעיקר משני סוגים:

1) מערכים הידברותיים ללימוד נושאים במדעי הניהול, נועדו למשתמשים שאינם מכירים כלל את השפה. מערכים אלה ניבנו כתרגיל בניתוח בעיה ויישום כתוכנה למחשב על ידי סטודנטים שאינם בעלי רקע מודגש במדעי המחשב.

2) אוספי פונקציות לניסוי אלגוריתמים-נועדו לשימוש על ידי המתכנת עצמו ומניחים ידיעת מה ב- APL . המטרה היתה תשובה מהירה לבעיות שהתעוררו תוך כדי מחקר בחקר ביצועים.

דוגמא למערכת הידברותית שנבנתה היא מערכת לניהול פרויקטים בשיטת הנתיב הקריטי. מערכת זו כוללת ארבע תת מערכות (ניתוח זמנים, ניתוח משאבים. חישוב עלויות, בסיס נתונים פשוט), ובנוסף פונקציות הסבר ומערכת עזר המאפשרת לכל המערכת לפעול בשטח עבודה מוגבל תוך שימוש בקבצים. המערכת נבנתה על ידי ארבעה צוותים שונים במשך שלושה סימסטרים עוקבים.

הלקחים שיוצגו מתייחסים לנקודת ראותו של מפתח התוכנה, אשר שיקולי היעילות שלו שונים, ואפילו מתנגשים, בשיקולי היעילות של מערכת המחשב. לדוגמא - שורת קידוד יכולה להיות יעילה מבחינת המשתמש אולם יקרה במשאבי מחשב בזמן בצועה. עם זאת, קיימים שיקולי יעילות שימוש במחשב המתבטאים בעת הביצוע באופן שהם משפיעים על תיכנון התוכנה.

העבודה נעשתה במערכת APL שבטכניון (APLSV, TSIO, מסופי 2741).

2. גורמים המשפיעים על מבנה תוכנה ב - APL.

2.1 סיווג הגורמים

הלקחים שיוצגו להלן נובעים לא רק מאופיה של APL כשפת תיכנות אלא גם מתכונותיה כמערכת ומהסביבה בה היא מופעלת. תכונות אלה ניתן לסווג באופן הבא:

APL כשפה .

דרך העבודה האינטרפרטיבית של APL .

שרותי APL כמערכת .

הגירסא המסוימת של APL (תוכנה וציוד) .

הקשר עם יתר חלקי המערכת ובעיקר גישה לקבצים ,

עבודה בסביבת שתף זמנים.

המסקנות לגבי APL כשפה וכמערכת תקפות למערכות APLSV, APL/cms ו - APL במחשב 5100. המסקנות הנובעות מאופי הגבלת המשאבים עשויות להשתנות בהתאם למערכת.

2.2. APL כשפת תיכנות

תכונותיה של APL כשפה מתבטאות בתפוקה, אירגון מערכת התוכנה, דרכי פתרון בעיות, תקני עבודה ואמצעי אבטחה (החשובים במיוחד במערכות הידברותיות).

בכלל, מתברר כי התפוקה ב- APL גבוהה מאד לעומת שפות אחרות, וכי מתאפשרות דרכי פתרון רבות וגמישות למיגוון רב של סוגי בעיות. עם זאת, מתעוררות מספר בעיות אופיניות.

להלן ידונו מספר תכונות בולטות של APL כשפה והודגש השפעתן.

1) מיגוון רחב של פונקציות יעילות - מיגוון זה מאפשר ל"בטא" רעיון תיכנותי באופן תמציתי ביותר בצורות רבות ושונות ולהתאים את התיכנות למגבלות משאבים (לחסוך במקום בזכרון או בזמן ריצה, לפי הצורך).

מצד שני, עלולה תכונה זו להפריע לקריאות התוכנה (ולכן לאחזקתה ושיפורה). כן עלול מתכנת לראות את התיכנות כאוסף של חידות - מהי הדרך "נאותה" ביותר להציג רעיון נתון. לשתי בעיות אלה השלכה על ההדרכה שיש לתת למתכנת ב- APL .

2) פעולות על גבי מערכים - מיגוון הפעולות האפשרי, לגבי מערך בודד ובין מערכים, נותן ל- APL אופי מתאים במיוחד לסוגים רבים של ישומים. דוגמאות שימוש:

עיבוד רשימות (List Processing) - נח במיוחד לתכניות דימוי.

טיפול במחרוזות תווים. נח לעיבוד טקסטים ולהפקת דוחות.

עידכון מספר רב של רשומות ושדות במספר מועט של שורות קידוד.

ניסוח נח של בעיות אופטימיזציה רב מימדיות.

3) תכונות אפיניות של פונקציות המשתמש - אין פונקציות פנימיות, כל שתי פונקציות הן חיצוניות זו לזו ועשויות להפעיל זו את זו. עולמה של פונקציה מושפע על ידי האופן בו הופעלה. לפונקציה לכל היותר שני פרמטרים והיא מטכפלת את הפרמטרים הממשיים בזמן הפעלתה.

השלכות של תכונות אלה: שימוש במשתנים גלובליים (המוכרים מחוץ לפונקציה המופעלת) להעברת נתונים בין פונקציות. צורך בקידוד מיוחד לפירוק פרמטרים מורכבים. מבנה מודולרי "טבעי" של התוכנה. אפשרות כתיבת משפטים בעלי משמעות המורכבים משמות פונקציות. כן יש השלכות לגבי תקני שמות ותיעוד, אלו ידונו בפרק 3.

4) קיימים ארבעה סוגי נתונים השונים מאד מבחינת תפיסת זכרון: לוגיים, תווים, ספרתיים שלמים, ספרתיים נקודה צפה. המעבר מסוג לסוג (פרט לתווים) עלול ליצור הפתעות מבחינת תפיסת מקום בזכרון ומבחינת מבנה הפלט (אם אינו מבוקר).

5) משתנים גלובליים ומקומיים - כל משתנה שהוצב בו ערך ממשיך להתקיים (עד בטולו), אלא אם כן הוגדר כמקומי בפונקציה. תכונה זו מחייבת לדאוג במפורש שלא ישארו משתנים מיותרים. כן רצויים תקני שמות לסימון משתנים גלובליים ולמניעת התנגשות בין תת מערכות.

6) אפשרות ביצוע מחרוזות כאילו היו בטויי שפה (פונקצית השפה ⍎). לתכונה זו השלכות נרחבות (ואף מפתיעות), להלן תתוארנה כמה מהן: ניתן לקבל כקלט ביטויים ולבצעם (לדוגמא: תנאי שליפה מתוך רשומה). ניתן לבנות מערכים כלליים (למשל מערך קליטה) באמצעות פונקציות פשוטות וטבלה המכילה מחרוזות לביצוע. שינוי המערך מתבטא בשינוי מחרוזת בטבלה ללא צורך בשינוי תיכנותי.

ניתן לכתוב בטויים המתיחסים למשתנים בעלי "שם כללי" וליצור את המשתנים לפני הביצוע. תכונה זו מאפשרת טיפול נוח בבסיסי נתונים.

כללית, אפשרות זו מרחיבה את מושג ה"נתון" כך שיכלול ביטויים הניתנים לביצוע.

להלן שתי דוגמאות: 1) תוכנית עידכון המתייחסת לרשומה בעלת מבנה כללי כאשר שורת קידוד אחת מבצעת פעולת עידכון כלשהיא (החלפת ערך, תוספת או הפחתה).
2) בתכנות דימוי מתבצעת הגדלת ערכים אקראיים על ידי "הפעלת" שורה בטבלת מחרוזות. המשתמש בוחר את מספר השורה. תוספת פונקציה או ביטוי כרוכה בהוספת שורה לטבלה, ללא שום שינוי בתכנית הדימוי עצמה.

2.3. דרך העבודה האינטרפרטיבית

לדרך העבודה האינטרפרטיבית שלש השלכות עיקריות:
1) תכונות שפה מסוימות מתאפשרות עקב אופן עבודה זה (בפרט: הגדרת משתנים תוך כדי ביצוע וביצוע מחרוזות כביטוי שפה).

2) ניתן לשנות פונקציות משתמש תוך כדי ביצוען. תכונה זו מאפשרת ניפוי מהיר של אוסף פונקציות על ידי ביצוע תיקונים מיד עם התגלות טעות והמשך העבודה מאותה שורת קידוד.

3) לביצוע שורת קידוד יש תקורה של אינטרפרטציה, כתוצאה, חזרה על שורת קידוד מספר רב של פעמים עלולה להשפיע בצורה חריפה על זמן הביצוע. עקב מגוון הפעולות האפשריות על מערכים ניתן להתגבר על בעיה זו ברב המקרים בהם אין משמעות לסדר החזרה על הלולאה. (לדוגמא: בעיבוד קבצים).
פתרון זה עלול להיות על חשבון מקום רב בזכרון.
בבעיות המצריכות תהליך חוזר בו הסדר משמעותי (לדוגמא: דימויים רבי אירועים, פתרונות של משוואות דיפרנציאליות, אופטימיזציות) פתרון בפעולה במקביל אינו ישים וכאשר חישוב שורה גוזל זמן נמוך לעומת פירושה עלול הפתרון ב- *APL* להיות לא יעיל. מבחינת המשתמש מתבטא הדבר בזמן תגובה רב ובהפרעות למהלך הפונקציות.

2.4. מערכת APL

מערכת APL מספקת למשתמש ליד המסוף ארבעה סוגי שירותים.

1) פקודות מערכת (SYSTEM COMMANDS) - מאפשרות לקבל את שטח העבודה הפעיל וספריות שטחי עבודה שמורים, מהוות כלי נוח ויעיל לשמירה, העתקה, גיבוי, שיחזור ואירגון אוספי פונקציות ומשתנים ולכן מאפשרות לטפל בקלות בארגונן של מערכות תוכנה מורכבות.

2) שירותי עריכת פונקציות מאפשרים בניה ושינוי מהירים של פונקציות.

3) אפשרויות מעקב - מאפשרות הדפסת תוצאת ביצוע שורות רצויות, עצירה במקום רצוי, הצגת ערכי משתנים בפונקציה שהופסק ביצועה.
שלושת סוגי השירותים הללו מאפשרים תפוקה גבוהה בבנית מערכות, בדיקתן ושינויין.

4) פונקציות ומשתני מערכת - מאפשרים למשתמש לשלוט על עצמים בשטח העבודה באופן מתוכנת. בפרט: אפשרות ביטול עצמים, הפיכת פונקציות למטריצות ולהיפך, אפשרות מדידת המשאבים (זכרון, זמן) המנוצלים ותכנות מהלך הריצה בהתאם לכך.

2.5. עבודה בשתף זמנים

עבודה בשתף זמנים מתאפיינת במגבלות המשאבים המוקצים למשתמש.
מגבלות אלה מתבטאות בזכרון מוקצה, זמן cpu (לפני ביצוע התערבות) וזמן תגובה.

המגבלה העיקרית בה נתקלנו היתה גודל שטח העבודה (כלומר, הזכרון הפעיל המוקצה למשתמש בזמן עבודה). תיכנון מבנה התוכנה התחשב באילוץ זה. תיכנון זה מתבטא במבנה הפונקציות, אופי התיכנות ודרכי העברת נתונים בין הפונקציות. הגבלת הזכרון מפריעה לבנית התוכנה בדרך שתחסוך זמן ריצה.

במערכות מסוימות (APL / cms) ניתן להגדיל מאד את הזכרון אולם הדבר עלול להתבטא בירידות זמן התגובה. בכל מקרה, רצוי לטפל בזהירות בסוגי פעולות ה"אוכלות זכרון" ובמערכים רבי מימדים.

2.6. גורמים הלוויים במערכת

ניתן לחלקם לשלוש קבוצות:

1) ציוד - בעיקר המסופים. מהירות המסוף וצורת הצגת הנתונים (מכונת כתיבה/צג) משפיעים מאד על תיכנון הדו-שיח וכמות הפלט. בעותק קשה המיגבלה העיקרית היא מהירות ההדפסה. בצג יש לאפשר שיחזור תוצאות שנמחקו. (מובן כי בעיות אלה אינן אפייניות דווקא ל- APL)

2) תוכנה - הגורם העיקרי היא אפשרות שליטת המשתמש על המשאבים המוקצים לו. במערכות APL בהן ניתן להגדיל את הזכרון, נוח הרבה יותר לעבוד אולם הדבר עלול לבוא על חשבון ירידת זמן התגובה והפיכתו לתלוי יותר בפעולות המבוצעות.

3) קשר עם יתר חלקי המערכת - הכוונה בעיקר לגישה לקבצים ובמידת מה לקלט מקורא כרטיסים ופלט במדפסת מהירה. ההשלכה העיקרית היא על כמות הנתונים שניתן לעבד (אפשרויות עיבוד נתונים).

3. מסקנות

ההשלכות המידיות של תכונות שונות של APL נידונו לעיל. בפרק זה תובאנה מספר מסקנות אשר לא בהכרח נובעות מתכונה יחידה.

3.1. סוגי בעיות "מתאימים" ו"לא מתאימים" לעבודה ב- APL

APL כשפה נוחה ומהירה מאד לפיתוח ישומים כמעט מכל סוג שהוא. לכאורה, אין השפה כוללת אפשרויות הקיימות בשפות אחרות, אולם עקב מיגוון הפונקציות והטיפול הנרחב במערכים נוח לבנות פונקציות מתאימות. מכיוון שפונקצית משתמש מתנהגת כפונקציה בשפה נובע כי ניתן ל"שכן" ב- APL תכונות רצויות.

דוגמאות: עיבוד רשומות (List Processing). טיפול במבנים (דוגמת Structure ב- PL/1).

ההשקעה בבנית פונקציות מיוחדות משתלמת עקב היותן "תפורות" לישום המסוים.

אי התאמה לבעיות מסוימות מתעוררת בזמן ריצה עקב מגבלות זכרון או עקב פעולות רבות החוזרות על עצמן. נקודת אי-הכדאיות לשימוש ב- APL תלויה במערכת.

עם זאת, בסוגי בעיות מסוימים, כגון פיתוח אלגוריתמים, עשוי השימוש ב- APL להיות כדאי בשלב מכין של בנית האלגוריתם לפני ישומו בשפה אחרת.

3.2. לימוד השפה והשימוש בה

קל ללמוד יסודות APL והאופי ההידברותי מקל על רכישת נסיון עבודה. עם זאת קיימות מספר בעיות בלימוד השימוש הנכון:

1) משתמש עלול ל"קפוא" ברמה מסוימת כי עקב גמישות השפה ניתן לפתור את רב הבעיות בהן הוא נתקל ללא שימוש בחלק ניכר מהפונקציות. תופעה זו עלולה להתבטא בתקורה אינטרפרטיבית גבוהה.

2) חלק ניכר מהמשתמשים מכיר שפות אחרות (בפרט FORTRAN ו- PL/1) ולהם נטיה לתכנת בדומה למקובל בשפות אלה. התוצאה כמו בבעיה הקודמת.

3) קיימת נטיה ללמוד חלק בלבד משירותי המערכת מכיוון שהם נראים למתחיל טפלים לשפה. התוצאה עלולה להיות ביזבוז זמן ניכר בהתעורר הצורך באירגון שטח העבודה או בשיחזור.

4) קימות תופעות מסוימות סמויות במידת מה. שתי דוגמאות: משתנה המוגדר בפונקציה אחת עלול להשתנות באופן סמוי באחרת מכוון שאין בשפה מושג של פונקציות פנימיות וחיצוניות). תוצאת ביטוי תלויה במציין האיבר הראשון בוקטור (0 או 1).

5) למרות ש APL שפה עילית, הרי עקב אילוצי המשאבים רצוי שהמשתמש יהיה מודע במידת מה לקורה במחשב. בפרט - לאופי האינטרפרטיבי של הביצוע ולהצגת משתנים בזכרון (לדוגמא: היחס בין המקום שתופסות הצגות 1.0 ו 1 הוא 1:64).

6) למשתמשים מתקדמים יותר נטיה לצרור בשורת קידוד מספר רב של פעולות שאינן קשורות. תופעה זו יוצרת תוכנה לא קריאה וגורמת תוספת פעולות.

הפתרון הנראה לכל הבעיות הללו הוא על ידי חינוך להימנע מהן. עקב הגמישות והגיוון נראה כי הדרך הטובה היא על ידי ריבוי דוגמאות באופן הדומה לא פחות לשפות מדוברות מאשר לשפות תיכנות.

3.3. תפוקה, אחזקה ושיפור, נוהלי עבודה

התפוקה ב- APL גבוהה במידה ניכרת מאשר בשפות עיליות אחרות הודות לשילוב בין עבודה הידברותית ותכונותיה של APL כשפה וכמערכת. הצימצום והמודולריות מורידים את זמן הקידוד, שרותי העריכה מאפשרים קליטה מהירה, המודולריות ושרותי העקיבה מאפשרים לבדוק פונקציות באופן עצמאי ושירותי המערכת מאפשרים שילוב מהיר של פונקציות ומשתנים במערכת.

תכונות אלה מאפשרות שינויים מהירים ביותר במערכת ומכאן נוחות רבה באחזקה ובשיפורים.

הקלות בה ניתן לשנות מערכת מחייבת תקני עבודה מסודרים וזהירים. בפרט: גיבוי שוטף של המערכת ורישום מידי של כל שינוי שנקלט. הימנעות מכך עלולה לעלות בזמן רב שידרש לשיחזור מערכת ששונתה או נמחקה עקב הוראה לא זהירה. בעיה מיוחדת היא נסייתו של שטח העבודה (הזכרון הפעיל) ל"הזדהם" במשתנים שנוצרו בשלב כלשהוא ובמחסניות (Stacks) של משתנים מקומיים בפונקציות שפעולתן הופסקה. גם בעיה זו מחייבת שיטת עבודה מסודרת, כמו למשל תקני שמות למשתנים הצריכים לשרוד בשטח העבודה, ו"ניקוי" לעתים מזומנות.

3.4. תיעוד

שלש מתכונותיה של APL משפיעות במיוחד על התיעוד הנדרש:

1) הצימצום בקידוד וריבוי הדרכים לקידוד בעיה נתונה.

2) העצמאות בהגדרה של פונקציות משתמש לעומת תלות סביבתן בזמן ביצוע בדרך בה הופעלו.

3) הקלות בה מופיעים, משתנים ונעלמים עצמים (פונקציות ומשתנים) בשטח העבודה.

כתוצאה מתכונות אלה רצויה שיטת תיעוד שתכלול:

1) תקנים לסימון המשתנים הגלובלים.

2) במערכות מורכבות - הקצאת שמות משתנים ופונקציות לחלקי המערכת השונים (מספיק לקבוע קידומות לשמות).

3) בכל פונקציה הערות שתכלולנה: תפקידה; הסביבה בה היא פועלת (משתנים גלובליים דרושים ופונקציות מופעלות); קלט: פרמטרים, משתנים גלובליים, מהמסוף, קבצים; פלט: מפורש, במשתנים גלובליים, מודפס, בקבצים; רשימת פונקציות המפעילות אותה.

4) תיעוד לקטעי פונקציות: תפקיד הקטע, קשירה לתיעוד כתוב.

5) הימנעות משורות קידוד מורכבות.

אופייני ל - APL שבפונקציות רבות עולה מספר שורות התיעוד על מספר שורות הביצוע.

3.5. שימוש בטכניקות תיכנות מתקדמות

APL נוחה לעבודה בתיכנות במבנים (Structured Programming) מכמה בחינות.

1) שורה בודדת עשויה לבטא רעיון תיכנותי מורכב למדי. כתוצאה מתקבל מבנה פונקציות בעל אופי "מקומי" ומעט הסתעפויות.

2) אופי פונקציות המשתמש מתאים באופן טבעי למבנה מודולרי.

3) APL מתאימה לשימוש כ"שפת על" (Pseudo Language) עקב היותה במקור שפה לניסוח אלגוריתמים. בחירה נאותה של שמות פונקציות מאפשרת להקנות לתוכנה דמיון לשפה מדוברת. יש כמה מגבלות לעבודה במבנים, שניתן להתגבר עליהן:

1) חוסר מבנים בסיסיים כגון DO ו IF ניתן לפתרון על ידי בנית פונקציות מתאימות.

2) תלות פעולת פונקצית משתמש בסביבה המפעילה - ניתנת לפתרון על ידי העברת נתונים בין פונקציות רק דרך פרמטרים ומשתנים גלובליים בשטח העבודה.

3.6. התגברות על מגבלות זכרון וזמן

מגבלות הזמן: נובעות מהתקורה האינטרפרטיבית. ניתן להתגבר עליהן על ידי ביצוע מספר מירבי של פעולות במקביל. לכך שני סוגים:

1) ביצוע פעולות חישוב מיותרות עלול לבטל את החסכון.

2) ביצוע פעולות במקביל כרוך בטיפול במערכים העלולים לתפוס מקום רב בזכרון ובכך יומרו מיגבלות זמן למיגבלות זכרון.

מגבלות מקום: המקום בשטח העבודה מנוצל לפונקציות, משתנים, ושטחי ביניים לביצוע פעולות (מוגדרים על ידי המשתמש או נצרכים על ידי APL עצמה). בפונקציות ניתן לחסוך בדרכים הבאות:

1) תיעוד והסברים מוחזקים בשטח עבודה שמור נפרד. בשטח העבודה הפעיל נמצאות רק פונקציות ללא תיעוד.

2) ניתן לפרק את המערכת לקבוצות פונקציות ומשתנים בשטחי עבודה שונים. החלקים יטענו בעת הצורך לשטח העבודה הפעיל באמצעות פקודות מערכת.

3) ניתן להחזיק פונקציות במחרוזות תווים בקבצים ולטעון אותן בעת הצורך לתוך שטח העבודה על ידי פונקציה מתאימה.

שתי השיטות האחרונות דורשות תיכנון ופירוק זהירים של המערכת.

במשתנים ניתן לחסוך בדרכים הבאות:

1) בחירה של צורות הצגה חסכוניות: שימוש בסיביות במידת האפשר, בשלמים נומריים במקום ערכים בנקודה צפה, אריזת מספר ערכים יחד על ידי צירוף (ENCODING).

2) חסכון במשתני ביניים: שימוש חוזר באותם משתנים, העברת נתונים בין פונקציות באמצעות משתנים גלובליים.

3.7. רגישות מערכות תוכנה

עקב היותה של APL מערכת "פתוחה" הניתנת לשינוי בנקל ועוברת שינויים דינמיים תוך כדי ריצה עשויה מערכת להיפגע עקב תיפעול לא נכון או עקב חריגה ממשאבים.

תיפעול לא נכון: הפעלת פונקציה לא בדרך הרגילה או קלט שגוי עלולים לגרום שגיאת ביצוע ופגיעה בנתונים קיימים כך שלמשתמש שאינו בקי ב- APL יהיה קשה להחזר את המצב לפני הטעות. תופעה זו מחייבת נקיטת אמצעי זהירות מיוחדים בעיקר במערכות הידברותיות.

טעות בפקודות מערכת עלולה לפגוע בתוכנה עצמה בשטח העבודה הפעיל. טעות כזו עשויה לקרות בנקל. הדרך להימנע מטעויות אלו היא ללמד את המשתמש שאינו יודע APL , פקודות מערכת. רצוי מאד להוסיף להוראות הפעלת המערכות הסבר כיצד לבצע שיחזור.

חריגה ממגבלות: קורית בזמן ביצוע עם מעבר מיגבלת זמן או תפיסת זכרון רב. ניתן להוסיף למערכות בדיקות מתוכנתות לשני סוגים. לזכרון - לפני ביצוע פעולות "מסוכנות" או בזמן הקלט, לזמן-בנקודות קבועות בתוכנה.

סיבה נוספת היא "הצפת" שטח העבודה עקב הצבת משתנים על ידי המשתמש או עקב הפסקות ריצה חוזרות ונשנות כתוצאה מתקלות. יש להדריך את המשתמש במערכת כיצד ל"נקות" את שטח העבודה.

4. סיכום

APL. היא שפה המתאימה למיגוון רב של סוגי בעיות. הזמן הנדרש להגשמת רעיון תיכנותי במחשב ב- APL נמוך במידה ניכרת מאשר בשפות אחרות.

תיכנון תוכנה ב APL מושפע משני שיקולים מנוגדים: הצורך להקטין את הזמן המושקע באינטרפרטציה ולעומתו הצורך להזהר שלא לחרוג ממגבלות זכרון.

דרך פתרון בעיות ב APL שונה מאשר בשפות עיליות אחרות כגון FORTRAN או PL/1 בעיקר עקב אפשרות ביצוע פעולות על מערכים שלמים.

APL מתאימה לעבודה בתיכנות במבנים.

כדי לאפשר פיתוח ואחזקה נוחים של מערכות תוכנה יש להקפיד ששורות הקידוד לא תהיינה מורכבות. רצוי לקבוע תקני שמות משתנים ופונקציות וחשוב להוסיף הערות לפונקציות.

רצוי מאד להדריך משתמש בתוכנה הכתובה ב - APL להכיר את יסודות השפה ואת השימוש בפקודות מערכת חשובות.

Auxiliary Chemistry Lessons at the Technion

E. Segall, M. Hackmeyer, Y. A. Sataty, and O. Marcovitch *

ABSTRACT

This paper describes APL functions that facilitate the writing of CAI tutorials and simulations of experiments. Using these functions a number of programs were written. These programs were used in the General Chemistry course at the Haifa Technion.

Introduction

An APL CAI program was initiated to stimulate students' interest in general chemistry. The students (the majority of whom are majoring in engineering) showed a lack of interest in the course and in chemistry in general. Therefore, a CAI program was initiated which allows the students to participate in the course on an individual basis. In this way a student could not only have the unique experience of using a computer terminal (which is the major stimulus at the first session), but he could also gain insights into experimentation, relationships between measurable quantities, and problem solving which would not be possible with lecture, laboratory, and recitation classes alone. Such insights could be of use in other fields of science as well.

In writing the package we tried to meet the following requirements:

1. The program should branch in response to numerical answers.

2. The student should be able to use the computer to calculate his answers.

3. The student should have some control over the running of a lesson, such as requesting an answer, skipping a question, and repeating a lesson.

* Computer Center, Department of Chemistry, Department of Chemistry, Department of Chemistry, respectively. Technion, Israel Institute of Technology.

4. The programming effort of the instructor should be minimized by providing functions for recurring operations, and default values for parameters that the instructor might want to change.

5. Writing a lesson should demand only a minimum of programming experience.

6. It should be possible to vary the response to an answer in accordance with the number of attempts at the answer; in other words, make branching dependent upon the number of times the branch-point is passed. The last problem was solved in a CAI package developed at Waterloo University, and we copied the pertinent functions from it.

When the student uses the computer to calculate the answer to some problem, he must be prevented from reading or, worse yet, changing, the values of any variables used in the lesson. Therefore the function that allows the student to calculate is generated during the running of the program, and all variable names appearing in the calling program are made local to the functions.

Availability: the described system has been submitted to to APL/PIE under the name "NUM CAI" for unrestricted distribution. APL/PIE is a shared APL library maintained at SUNY at Binghamton, NY.

Appendix I lists the principal functions used. Appendix II demonstrates the writing and use of a lesson (English version).

Set Up

Six IBM 2741 terminals were set up in a room next to the general chemistry laboratory. During each four hour laboratory meeting, a graduate assistant who was in charge of the computer work brought students into the terminal room five or six at a time to work on a relevant program. The terminal and laboratory sessions were made to coincide because otherwise there would be considerable difficulty in scheduling the students to use the terminals. It turned out that it was not as difficult as first anticipated to bring students from the laboratory to the terminal room mainly because they work in pairs in the laboratory. Each terminal session lasts about twenty minutes.

The supervisor (and, occasionally, each of the programmers)

watches the students in the terminal rooms and notes which parts of the programs are the most difficult, the least difficult, the most interesting, and the least interesting, and where there are unique problems. This information is relayed to the programmers who modify the program during the week that it is being used.

When a student finishes a terminal session, he takes the output with him and returns to the laboratory to complete his experiment. He attaches the computer output to his laboratory report which is graded and returned to him the following week. (The programs also have an internal tabulation system by which the student's performance at the computer terminal is analyzed).

Format of Programs

Most of the programs are designed to be simulations of the experiments which the student is performing in the laboratory. However the programs can penetrate much deeper into all aspects of the of the experiments and they also can help the student to analyze his results on the spot. For example, there is a program on the phase diagrams of pure substances to augment an experiment on the vapor pressures of pure liquids and solids. The program simulates three experiments; the vapor pressure of the pure liquid as a function of temperature, the vapor pressure of the pure solid as a function of temperature, and the variation of the melting point with pressure. After each of these three experiments, the program helps the student to analyze his data, to find the mathematical relationships between pressure and temperature (using IBM-supplied plotting functions), and to deduce all kinds of additional information from these relationships such as heats of vaporization, boiling and freezing points, and the difference in density between the liquid and the solid. At the end, the whole phase diagram is drawn, based on the three relationships which the student has deduced, and he is asked questions based on this phase diagram, such as where the triple point is.

The programs write in Hebrew using a Hebrew printing element. Since Hebrew is read from right to left and the terminal prints from left to right, the terminal prints the end of a line first andthe beginning of a line last. Another problem of the Hebrew printing element is that the mathematical symbols are not in the correct positions for APL. The proper translation to APL symbols is performed by a pertinent function.

Another type of program is used to help the student to learn how to solve problems. Each of these programs contains from five

to seven problems progressing from least difficult to most difficult. In this type there is more effort used to anticipate the kinds of mistakes that the students will make and to find the specific mistakes of an individual student.

In both types, chemical substances are chosen at random from a list which can easily be modified by those who are not familiar with computer programming. Numerical values in problems are also chosen at random within predetermined limits. Thus each student gets different problems, or, if a student wants to repeat a terminal session of a simulation of an experiment, he gets a different chemical substance for his experiment and different chemical substances and numbers in his numerical problems.

Interaction

Each program contains text, questions which can be answered yes or no, and questions which require numerical answers. If a student is asked a question which requires a yes or no answer, he answers Y or N (or the corresponding Hebrew letters). If he is correct he receives a message so indicating. Otherwise, he gets a message indicating that he is not correct, followed by eitheranother yes-no type question to try to discover his difficulty, or an explanation. if the student is asked a question which requires a numerical answer to a problem, he has three tries to get it right. During each try he has six options. He can use the terminal as a calculator (by sending the word DESK) , he can work out the answer without using the terminal (such as with a pocket calculator or by hand), he can ask for help by sending the word HELP, he can skip the problem entirely, he can skip the problem but receive the correct answer, or he can end the terminal session and receive a tabulation of his results. If the student gets the correct answer on the first try, he is given a very encouraging personal message (chosen at random from a list) followed by his first name. If the student gets the wrong answer, his answer is checked against a list of typical wrong answers, and, if agreement is found, the appropriate message is written explaining to him what he did wrong. If the student's wrong answer is not on the list, a series of yes-no questions are given to him to locate his difficulties. If the student gets the correct answer on the second try, he receives a moderately encouraging personal message, followed by his first name. If the student does not get the correct answer on the third try, he is given the correct answer and the program continues. If a student misses three questions in a row, he is given a short lecture (chosen at random from a list) on the appropriateness of studying before coming to class and the program then continues.

The tabulation of the student's results contains four entries for each problem. The entries are the number of tries, if he obtained the correct answer, whether or not he skipped the problem, and whether or not he asked for the correct answer to the problem.

Programs

The programs are in the process of being written. Those that have been completed incorporate the following topics:

1. Molecular weight, equivalent weight molarity and normality *
2. PH
3. Phase diagrams
4. Freezing points of binary solutions
5. Real gases
6. Ideal gases
7. Electrochemistry
8. Freezing point depression and boiling point elevation
9. Chemical Kinetics
10. Organic nomenclature

Results

The CAI program has been incorporated into the general chemistry curriculum, and the majority of students have expressed enthusiam towards it. Almost all of the students were able to progress through the programs without excessive difficulties. Students who missed their turn because of computer breakdowns or other reasons usually returned later to work at the terminals. Most of the students agreed that they got additional insights into the material by using the terminals. No attempt was made to use control groups to test the effectiveness of CAI since this had been done before.**

---

* A translated and modified version of CBIS CH020 SOL2 from:
Computer Based Instructional Systems Inc,
211, West Market Street
San Antonio, Texas 78205
The original program was written in the basic language.

** See, for example, S. Castleberry, G.H. Culp, and J.J. Lagowski, J. Chem. Ed. 50 (1973) 469.

APPENDIX I
On-Line Documentation

DESCRIBE

THIS WORKSPACE CONTAINS FUNCTIONS THAT FACILITATE WRITING OF COMPUTER-AIDED EXERCISES, PRIMARILY WITH NUMERICAL ANSWERS.

THE FOLLOWING FUNCTIONS ARE USED IN THE PREPARATION OF TABULAR DATA AND SHOULD NOT APPEAR IN THE STUDENT'S WORK SPACE (GROUP: "TABLEGRP"):

- CREATABLE - CREATES A FUNCTION
- TENTRIES - ENTERS DATA INTO A FUNCTION CREATED BY CREATABLE
- TADEL - DELETES LINES FROM A FUNCTION FORMED BY CREATABLE.

THE FOLLOWING FUNCTIONS ARE GENERALLY USED DURING RUNNING (GROUP: "RUNGRP"):

- INIT - INITIALIZES A LESSON
- COR - CHECKS FOR CORRECT ANSWERS, AND KEEPS TRACK OF STUDENT PERFORMANCE
- SEQ - PERMITS BRANCHING DEPENDENT UPON THE NUMBER OF TIMES A POINT IN THE PROGRAM IS PASSED
- READ, ΔALC - NEEDED BY COR
- DESK - CALLED BY STUDENT TO USE APL AS DESK CALCULATOR.

THE FOLLOWING FUNCTIONS MAY BE USED IF CONVENIENT (GROUP: "OPTGRP"):

- NUM - BRANCHES IN RESPONSE TO NUMERICAL INPUT
- TXT - BRANCHES IN RESPONSE TO CHARACTER INPUT
- COMS - RETURNS SPECIFIED PART OF CHARACTER VECTOR(I.E. TEXT)
- ΔSTOP - PRINTS OUT STUDENT'S PERFORMANCE. CALLS "ΔEXLES" TO EXPUNGE VARIABLES STARTING WITH "Δ".
- THEN, ELSE - SIMILAR TO PL/1 OR ALGOL IF- THEN-ELSE
- RAND - PRODUCES RANDOM VALUES WITHIN A SPECIFIED RANGE
- YGO, NGO - BRANCHES RESPONDING TO STUDENT'S "YES" OR "NO" ANSWERS. INSTRUCTIONS FOR USE OF THE FUNCTIONS ARE

OBTAINED BY TYPING THE NAME OF THE FUNCTION APPENDED BY "HOW".

<u>RESERVED NAMES</u>

- NO NAMES STARTING WITH "Δ" SHOULD BE USED EXCEPT ACCORDING TO INSTRUCTIONS.
- NO NAMES SHOULD HAVE "A̲" (UNDERLINED "A") AS SECOND CHARACTER.
- RESERVED NAMES OF INTEREST TO THE INSTRUCTOR ARE DESCRIBED IN "INITHOW".
- ALL GLOBAL VARIABLES ARE LISTED IN "GLOBALVARS".

ALL DOCUMENTATION IS GROUPED UNDER "DOCUMENTATION".

AUTHOR: ELAZAR SEGALL, TECHNION, ISRAEL INST. OF TECHNOLOGY, HAIFA, ISRAEL. LAST REVISED: MAY 1976. UNRESTRICTED DISTRIBUTION.

APPENDIX II

This appendix demonstrates the stages of writing a lesson: Creating a table, entering two items, and checking the table. Then the beginning of a lesson is shown, and a trial run. Parts in normal type are not part of an APL session. User's input is underlined.

```
      CREATABLE 'T1'          (Creating a table named "T1")
ENTER COLUMN WIDTH: 33
ENTER VARIABLE NAME OR LINEFEED: SUB
ENTER VARIABLE NAME OR LINEFEED: KS
ENTER VARIABLE NAME OR LINEFEED:
PLEASE TYPE A ZERO (0) UNDER EACH NUMERICAL VARIABLE AND A ONE (1)
      UNDER EACH CHARACTER VARIABLE
SUB                                 KS
1                                   0
ENTER CODE FOR T1: C
T1

      TENTRIES 'T1'        (entering data into a table)
SUB                                 : TRICHLOROACETIC ACID (CL3CCOOH)
KS                                  : 1.47E¯3
PLEASE TYPE "Y" TO ENTER MORE DATA.
Y
SUB                                 : HYDROFLUORIC ACID (HF)
KS                                  : 6.76E¯4
PLEASE TYPE "Y" TO ENTER MORE DATA.

T1

      'C' T1 'P'   (Checking the table by printing it out)
 NO  SUB                                KS

  1  TRICHLOROACETIC ACID (CL3CCOOH)   1.47E¯3

  2  HYDROFLUORIC ACID (HF)            6.76E¯4
```

Beginning of a program:

```
    ∇ PH
[1]   INIT
[2]   →4
[3]  STOP:→ΔSTOP
[4]   ''
[5]   'C' TA 'R'
[6]   ⍎'C' TA 'O'
[7]   CONC←RAND 0.001 0.1
[8]   'WHAT''S THE PH OF A ',(⍕CONC),' MOLAR SOLUTION OF '
[9]   SUB,', WHERE THE KA OF ',SUB,' IS ',(⍕KS),'?'
[10]  ''
[11]  HELP←SUB,' IS A WEAK ACID. HX = H(+) +X(-), KA=[H(+)]×[X(-
      )]÷[HX]'
[12]  ANSW1←(KS×CONC)*0.5
[13]  ANSW←-10⍟ANSW1
[14] Q1:→Q2 COR ANSW,(0.02×ANSW),(0.01×ANSW)
[15] B1:→SEQ B1+ 1 7
[16]  →(0.02>|(ΔA-ANSW1)÷ANSW1)/A1
[17]  'YOUR ANSWER IS NOT CORRECT.'
[18]  HELP
[19]  'PLEASE TRY AGAIN.'
    ∇
```

COMMENTS TO pH:

Line	1	"INIT" greets the student, asks his name, sets initial and default values etc..
	2	Go to line 4.
	3	"Delta STOP" prints statistics and returns a zero, stopping the run.
	4	Line numbers of acids in table T1. As "S" means "select", "'C' T1 INS" will be used to obtain data from table T1 in problems involving acids.
	5	Skip a line.
	6	Read a line from T1, giving values to SUB and KS.
	7	Set "CONC" somewhere between 0.001 and 0.1.
	8-9	Text is printed.
	11-13	Text and numerical values are stored under names "HELP", "ANSW1", and "ANSW".
	10	If answer is within 0.02 of ANSW without using the computer, or within 0.01 of ANSW when using the computer to calculate the answer, branch to Q2. "COR" asks the student for input and analyses it, and stores it under "delta A". It compliments the student for correct answers and updates certain statistics.
	15	After first unsuccessful attempt, go to next line; after second, to line 21. (Line number at label B1 plus 1, and plus 6.)
	16	If the answer is close to the H-ion concentration (instead of the pH), go to the line labelled A1.
	17	The line is printed.
	18	Text stored in "HELP" is printed.

Beginning of the lesson:

```
      PH
GOOD AFTERNOON
WHAT IS YOUR FIRST NAME?
ELI
THIS MODULE CONTAINS EXERCISES IN PH PROBLEMS.

WHAT'S THE PH OF A 0.0653 MOLAR SOLUTION OF
HYDRAZOIC ACID (HN3) (KA=1.91E¯5)?
HELP
HYDRAZOIC ACID (HN3) IS A WEAK ACID. HX = H(+) +X(-), KA=[H(+)]×[X
      (-)]÷[HX]
DESK
WHEN YOU KNOW THE ANSWER, TYPE "RETURN".
⎕:
      (.0653×1.91E¯5)*0.5
0.00111679452
⎕:
      RETURN
TYPE YOUR ANSWER, PLEASE.
.00111679
THAT'S THE H-ION CONCENTRATION. WHAT IS THE PH?
2.9520
KEEP IT UP! NOW YOU'VE GOT IT ,ELI

WHAT IS THE PH OF A 0.0555 MOLAR SOLUTION OF
ETHYLAMINE (CH3CH2NH2), (KB=0.00047)?
PLEASE
ANSWER IS 11.70819542


ELI     STUDENT NUMBER: NONE

PROBLEM: 1 2
CORRECT: 1 0
ATTEMPS: 2 0
GIVEN  : 0 1
SKIPPED: 0 0
```

קבוצת דיון מס׳ 6:

"מחשבים בחינוך"

יו"ר: ד"ר י. פלס

י. פלס

מ. רוס

ל. אוסין

נושאים בתחום המחשבים בחינוך, בפיתוח במרכז לטכנולוגיה חינוכית.

ד"ר י. פלס, פרופ' א. פנואלי, חנה גולדמן,
א. אילון, פ. שליטא, י. ענברי.

מבוא.

המרכז לטכנולוגיה חינוכית נוסד בשנת 1970 על ידי קרן רוטשילד, ונטל על עצמו, כאחת ממשימותיו העיקריות, את חקר השימושים האפשריים של המחשב במערכת החינוך בישראל.

במשך שש שנות פעילותו טיפל המרכז לטכנולוגיה חינוכית (מט"ח) בפיתוחם וקידומם של 4 פרוייקטים בתחום שימושי המחשב בחינוך.

ואלה הם:

א. הוראת "יסודות תורת המחשב" בבית-הספר העל-יסודי. (1)

ב. "מחשבים ב"-פיתוח נושאי-לימוד במדעי-המחשב בכתות י"א - י"ב של בית-הספר העל-יסודי. (2)

ג. "אשתולי" (אבחון שוטף של תוצאות למידה)-פותח בשיתוף עם המרכז לטלוויזיה לימודית. (3)

ד. "תואיים" (תרגול ואימון)-הוראת חשבון בביה"ס היסודי בעזרת מחשב.

הראשון בהם - הוראת "יסודות תורת המחשב" בכתה י' של בית הספר התיכון- הפך לתוכנית מוכרת במערכת החינוך, ותלמידים הלומדים במסגרתו מוגשים למבחני-הבגרות בנושא זה.

במאמר זה נסקור בקצרה את שלושת הפרוייקטים הראשונים, ונתאר בהרחבה את האחרון מבחינה כרונולוגית-את פרוייקט תוא"ם.

---

(1) "הוראת יסודות תורת המחשב" בבתי-הספר העל-יסודיים-ד"ר י. פלס, פנחס שליטא -"מעשה חושב"-עלון איל"א-מס' 2, אפריל 1976.

(2) "הוראת תורת המחשב, שלב מתקדם"-א. אילון - "מעשה חושב", כנ"ל.

(3) "אשתולי" (אבחון שוטף של תוצאות למידה), מבחנים ובדיקתם באמצעות מחשב כחלק אינטגרלי של תהליך הלמידה-שולה בין ויצחק ענברי (פרוייקט מחקר של המרכז לטלוויזיה לימודית והמרכז לטכנולוגיה חינוכית) - "מעשה חושב", כנ"ל

הפרוייקט הראשון - "הוראת יסודות תורות המחשב" בבתי הספר העל-יסודיים-
מכוון לכתות י' של בתי-ספר אלה, אולם ניתן ללמדו גם בכתה י"א.

בשלבי התכנון של פרוייקט זה, שהחל בשנת 1970, נבחנו שאלות הקשורות באוכלוסיה הלומדים, ונקבעה התאמתו של הנושא לכתה י'. הוחלט כמו-כן לתת את הקורס לכל התלמידים, ולא לקבוצות נבחרות. בעזרתה של וועדה מיעצת (4), נקבעה תוכנית להכשרת מורים יסודית וממושכת, נקבעה מסגרת סדירה של שלוש שעות בשבוע להוראת הנושא בכתה, נבחרה שפת-תכנות מתאימה למטרות הקורס - שפת BASIC, ונדונו דרכים לעבודה אינטראקטיבית של התלמיד בעזרת מסופים, וכן להרצת תוכניות בכרטיסים. בהתאם לכך החליטה הוועדה גם על רכישת הציוד המתאים. כמו-כן נדון ונקבע גם הרכב תוכנית הלימודים.

בכל הסוגיות הללו נבדקו ונמצאו הדרכים המתאימות לתנאי בית-הספר התיכון בישראל.

לאחר סיום הכשרתה של קבוצת המורים הראשונה, הוחל בקורס הראשון, שהתקיים בשנה"ל תשל"ב. לקראת תחילת שנת לימודים זו נכתב ספר-לימוד מיוחד להוראת הנושא בבית-הספר. כמו-כן פותח ה"מח"ל" (המחשב הלימודי) לצורך הדגמה, תרגול ולימוד מבנה המחשב ושפת-מכונה.

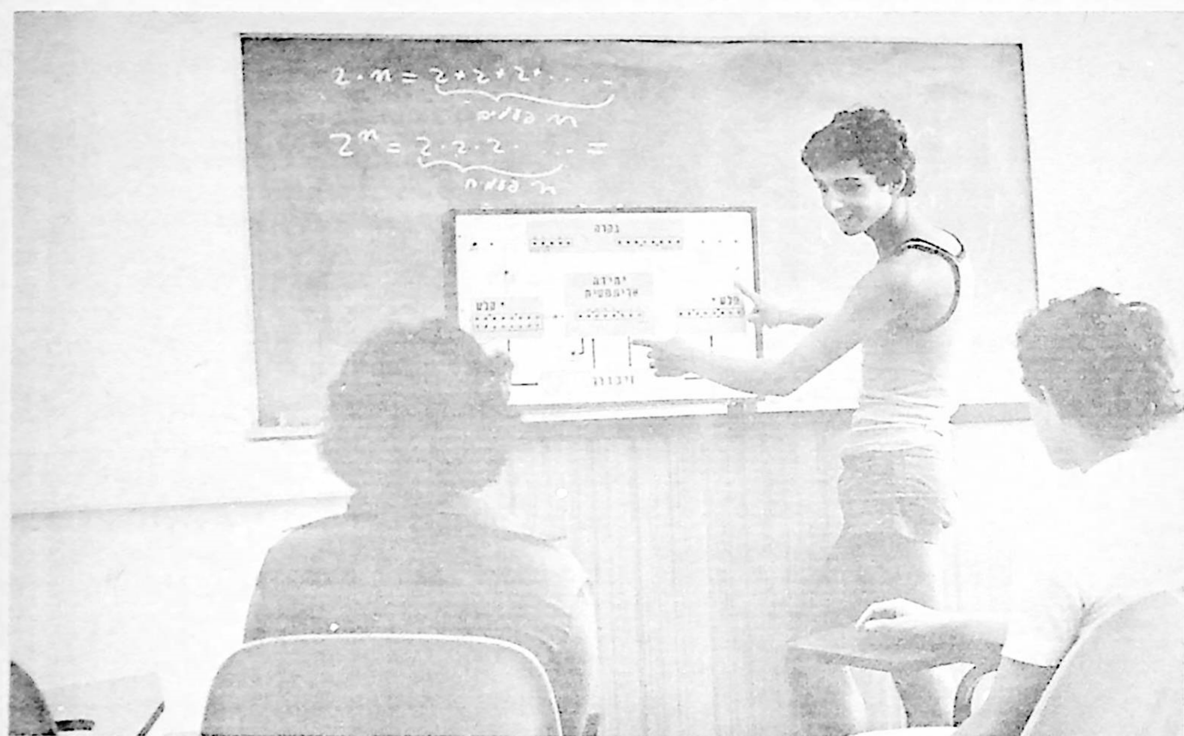
בכל אחד משלבי הניסוי נערך מחקר על ציפיות, עמדות, והישגי התלמידים. מתוך מחקר זה מתברר שתלמידי כל המגמות מגלים ענין רב בנושא, ובעיקר תלמידים המתעתדים ללמוד במגמה ריאלית או בנתיב הטכנולוגי.

זה עתה הסתיימה השנה הרביעית לניסוי, ובמשך ארבע שנים אלה למדו את הנושא כ-6000 תלמיד, והוכשרו להוראת המקצוע למעלה מ-80 מורים, בקורסים של המרכז לטכנולוגיה חינוכית.

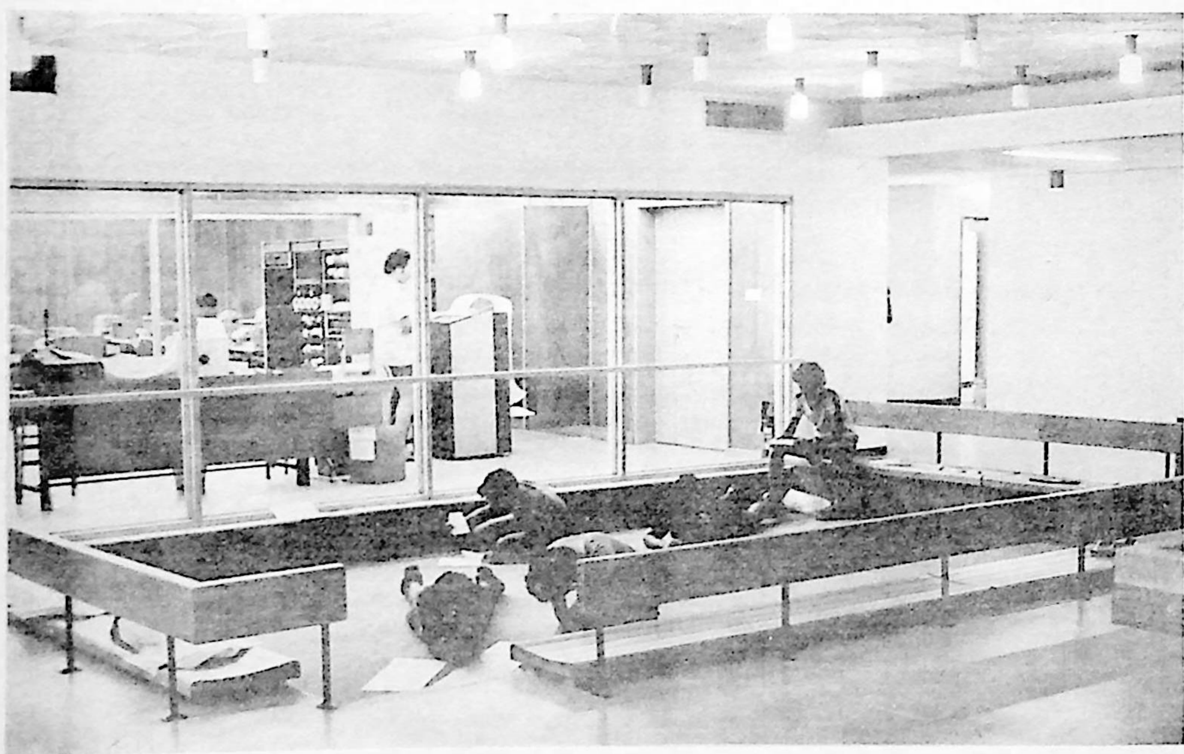
הפרוייקט השני - המשך לימודי המחשב בכתות י"א - י"ב (מחשבים ב').

במחקר-מעקב שנערך בסיום שנת הלימודים תשל"ה, בין מסיימי כיתות י"א וי"ב, שלמדו את הנושא "יסודות תורת המחשב" בכתה י', לשאלה:"האם רצית להמשיך וללמוד בנושא המחשבים", היתה תשובתם של 62% מהתלמידים (79% מבין הבוגרים במגמה ריאלית ו-43% מבין הבוגרים במגמה הומנית)-חיובית. תוצאות דומות התקבלו במחקרים שנערכו גם בשנים קודמות.
לאור זאת נתבקש המרכז לטכנולוגיה חינוכית ע"י משרד החינוך והתרבות, לפתח קורסים נוספים בתחום זה.

(4) בוועדה המיעצת השתתפו: פרופ' י. גיליס-מכון וייצמן למדע (יו"ר הוועדה), פרופ' א. גינזבורג-הטכניון, מר ע. בן-כוכב - אוניברסיטת בר-אילן. נציגי משרד החינוך היו: מר א. מרקוס, המפקח-המרכז להוראת המתימטיקה, וד"ר כרמי יוגב, יו"ר הוועדה לחינוך על-יסודי במזכירות הפדגוגית.



$2 \cdot n = 2+2+2+\dots$
n פעמים
$2^n = 2 \cdot 2 \cdot 2 \cdot \dots =$
n פעמים
בקרה
יחידה אריתמטית
קלט
פלט
זיכרון



בוועדה הארצית המייעצת להוראת מדעי המחשב בבתי הספר העל-יסודיים התקיימו דיונים בנושא, והוגדרו המטרות לאורן יש לבנות את תוכנית הלימודים. מטרות אלה הן:

1. העמקת הידע במדעי המחשב.

2. שימוש במחשב לשם הרחבת הידע בתחומים נוספים.

3. קידום יכולת התלמיד לעסוק ב"פתרון בעיות" במובן רחב יותר.

4. לעודד את התלמיד לדרכי חשיבה חדשות ובלתי שגרתיות.

5. לאמן את התלמיד המוכשר ביישומים מתקדמים של המחשב, וללמדו לכתוב תוכניות מחשב ברמה נאותה.

הוועדה המיעצת מינתה ועדת-משנה אשר תקבע את תוכנית הלימודים בנושא, ותבחן את כל האספקטים של פיתוח הנושא, כולל כתיבת ספר לימוד והכנת סגל-ההוראה.

הנושאים שנמצאו מתאימים בכתות אלה הם:

א. לימוד שפת מכונה ושפות תכנות.

ב. יישומים במתימטיקה, פיסיקה וטכנולוגיה.

ג. יישומים בתורת ההסתברות.

ד. לוגיקה.

ה. עיבוד נתונים.

ו. חקר ביצועים.

ז. תורת המספרים.

גם בשלב לימוד זה יהיה לפעילות התלמידים מחוץ לכתה חלק נכבד, והיא תתבסס בעיקר על הרצת תוכניות מחשב בכרטיסים, פעילות אינטראקטיבית במסופים וכתיבת עבודות ע"י התלמידים.

"מחשבים ב'" - לאחר גיבושה - תהפוך אף היא לתוכנית לימודים מוכרת למבחני הבגרות והגמר.

עדיין קיימות שאלות הדורשות ליבון, הן בתחום הלוגיסטיקה של הנושא, והן בתחום גיוסם והכשרתם של המורים ובתחום הדידקטי, אולם למרות זאת ברור הצורך בהשתרשותו והרחבתו של המקצוע "מדעי המחשב" במערכת החינוך העל-יסודי.

<u>הפרוייקט השלישי-"אשתולי"</u> (ראשי תיבות של "אבחון שוטף של תוצאות למידה"), הוא פרוייקט נסוי שנערך בשיתוף עם המרכז לטלוויזיה הלימודית, במטרה לקדם למידה יחידנית בעזרת המחשב.
הרעיון הוא ליצור מערכת מחשב אשר: תבדוק מבחנים תקופתיים של תלמידים; תאפשר למורה למצוא את נקודות התורפה של הכתה בכלל ושל התלמיד בפרט ולנסות לתת לו את האמצעים כדי להתגבר על כך; תאתר לתלמיד הבודד את נקודות התורפה בלמידתו, ותנחה אותו להשתמש בדפי עבודה שיעזרו לו להתגבר על קשיים.

השיטה מורכבת:

- ממערכת מבדקי איבחון.

- מעיבוד ממוכן של המבדקים והפקת דוחות למורה וגם לתלמיד.

- מדפי עבודה "יחידניים" המתמקדים על סוגי שגיאות אופיניות.

השיטה מתכוונת להשיג עידוד לימודו של התלמיד בדרך של הכוונתו תוך כדי לימודיו. למורה היא נותנת תמונה על הישגי התלמיד, בלי העול המכביד של בדיקת מבחנים.

הנושא של הניסוי הפעם היה: שיעורי השפה האנגלית, הניתנים ע"י הטלוויזיה הלימודית. הטכניקה של אשתו"ל מורכבת מפעילויות המבוססות על מבחנים "רבי-ברירה": פעילויות הכנה, הגשת הבחינה, בדיקתה, הגדרת המבחן למערכת הממוכנת. עיבוד התשובות במחשב והחזרת הדוחות לכתה.

פותחו "כרטיסי-מבחן" שאינם כרטיסים מנוקבים, אלא מסומנים ע"י התלמידים עצמם ידנית, ונקראים ע"י קוראת- כרטיסים אופטית. כמו-כן פותחו דוחות לתלמיד ולמורה וכן דוחות סטטיסטיים מסכמים.

שאלות נוספות, אשר התעוררו תוך מהלך הניסוי היו בעיקרן:

- נושא הרחבת השימוש במחשב לבדיקת מבחנים.

- נושא "בנק המבחנים" והפוטנציאל שלו.

- בעיות העלות והכדאיות של תפעול מערכת בדיקת מבחנים.

- בעיות משלוח המבחנים אל מרכזי המחשבים ומהם אל המורה.

- בעית ה"העתקות" ומניעתן.

בקנה-מידה גדול יהיה מחירו של מבחן בודד כ-50 אגורות, ושל מבחן כיתתי - 15 עד 16 לירות - מעידים כי הרעיון אינו יקר מדי, ואילו התועלה הטמונה בפיתוח המערך בקנה מידה רחב, לתלמיד ולמורה כאחת, מזמינה את הגשמתו.

__הפרוייקט הרביעי-תוא"ם (תרגול ואמון-Drill and Practice).__

פרוייקט זה מבוסס על הקוריקולום של כתות א' - ו' של בית הספר היסודי, ויכול לשמש תלמידי כתות אלה וכן תלמידי כתות גבוהות יותר הזקוקים לחזוק ותרגול.

בשלב הנוכחי מטפל הפרוייקט בנושא החשבון. הלימוד בו מבוסס על תרגול ואמון בעזרה מערכות תרגילים מהירות ומתוכננות היטב, למשך עשר עד חמש עשרה דקות ליום, על פי תוכניה לימודים בחשבון הידועה בשם Mathematical Strands, שפותחה על ידי חברת Computer Curriculum Corporation, בעקבות מחקר שנערך ע"י פרופ' פ. סופס (P. Suppes) ועוזריו, באוניברסיטת סטנפורד. זוהי גירסה של פרוייקט חינוכי שפותח באוניברסיטה זו, אולם גירסה זו הותאמה לתוכנית הלימודים הנהוגה בישראל, לציוד מודרני יותר וזול יותר מאשר בניסוי המקורי, וכן הוכנסו בה שינויים ותוספות שפותחו במט"ח.

Mathematical Strands בנויה מסדרת מסלולי לימוד (strands) בנושאי חשבון מגוונים כגון: מושגים על מספרים, חיבור אופקי, חיסור אופקי, חיבור וחיסור במאונך, משוואות, כפל אופקי, חוקי האריתמטיקה, כפל במאונך, חילוק, שברים פשוטים, שברים עשרוניים, מספרים שליליים, וכו'.

בכל מסלול מוצגת בפני התלמיד סידרה של שאלות בעלות קושי מדורג. שאלות אלה אינן מוכנות ומאוחסנות בזכרון המחשב, אלא מיוצרות על ידי המחשב בטכניקת המספרים האקראיים, אולם על-פי אלגוריתמים בהם הוכנסו הפרמטרים המתאימים לכל סוג שאלות הרוטינה המתאימה לא רק יוצרת את הערכים המספריים של התרגיל ואת המידע הדרוש לעריכתו, אלא גם מחשבת את התשובה הנכונה כדי להשוותה עם תגובתו של התלמיד. המחשב גם דואג להחלפת השאלות ולמעבר ממסלול לימודי אחד לשני, תוך שמירה על רמה אחידה של התרגילים במסלולים השונים, ותוך קידומו, או הורדתו, של התלמיד לרמה המתאימה לו על-פי מידת הצלחתו.

השינויים והתוספות שהוכנסו על ידינו בתוכנית המקורית כוללים לדוגמה, strand נוסף של שאלות המנוסחות במילים (להלן: שאלות מילוליות), דוח על הקושי היחסי של כל תרגיל וכן דוח אוטומטי על תלמידים שהתקדמותם נעצרה במסלול כלשהו.

הכנון, המערכת והתכנות בוצעו ע"י חברת "מיני-מערכות" - ישראל.

להלן נתאר: א. את הציוד (Hardware) המשמש את הפרוייקט.

ב. את שלב התכנון של המערכת.

ג. את התוכניה (Software) בהן היא משתמשת.

ד. את הקבצים ההיצוניים המוקמים על ידה.

ה. את אופציות הפיתוח הנפתחות בפניו.

__תאור כללי של מערכת תוא"ם.__

נתחיל במרכיבי הציוד (Hardware): מחשב DEC PDP11/10 עם זכרון בעל 24K, ו-Multiplexor בעל 16 ערוצים. ברצותנו נוכל להוסיף Multiplexor נוסף ואז תוכל המערכת לשרת 32 מסופים ללא ירידה משמעותית בביצוע וללא תוספת זכרון.

מרכיב נוסף של המערכת הם שני כונני תקליטונים (Floppy disks), המשמשים לאגירת נתונים ואחסון מערכת התוכניות.

המרכיב האחרון - 16 מסופי CRT, כל אחד בעל 24×80 מקומות להצגת תוים (אותיות וספרות), ומצוייד באופציה להצבת מצביע (cursor) במקום הרצוי. המסופים מתוצרת אלביט, ישראל, פועלים בקצב של 2400 סביות לשניה (bauds).

מבחינת ה- software מאורגנת היחידה כמערכת רב-תיכנותית - (multiprogramming system) כל מסוף במערכת מהווה משימה (task), וכל משימה מזוהה ע"י מספר המסוף. אחת המשימות מוכרזת כמשימת פיקוד (proctor task) ומוטלים עליה תפקידים מיוחדים ואילו האחרות מוגבלות להצגת תרגילים ובדיקת נכונות תגובות התלמיד. כל המשימות משתמשות באותן התוכניות, ולכל משימה איזור עבודה שלה בזיכרון, בה מאוחסנים הערכים המיוחדים לה.

התוכניות המשרתות את המערכת נחלקות לשלשה סוגים, בהתאם לשלוש רמות קדימות המוכתבות על ידי התוכנה.

תוכניות רמת הפיקוח(supervisor level) מהוות את הרמה הגבוהה - level 3, רמת הקדימות שמתחתה - level 2 - מכילה תוכניות שבהן מעטות פעולות העיבוד בין הקלט והפלט, ועל כן משהות רק במעט את פעולת התוכניות בעלות הקדימות הנמוכה ביותר - level 1. ברמה זו כלולות התוכניות המפעילות את המשימות.

מהי משימה? זוהי לולאה תמידית בעלת המחזור הבא:

1. הדפסת פעולה במסוף.
2. קליטה ופיענוח של הוראה תקינה.
3. קריאה לתוכנית המתאימה לביצוע ההוראה שהודפסה.

לאחר ביצוע התפקיד הנדרש, המשימה חוזרת לנקודה 1.

התוכניות המהוות את level 1 מבצעות, למעשה, את רוב משימות המערכת, ומכונות בשם המצביע על פעולתן. לדוגמה-ההוראה Proctor מכריזה על אחד המסופים כמסוף-פיקוד. לאחר הכרזה זו מותרות כל ההוראות פרט להוראה Drill, עליה ידובר בהמשך, רק ממסוף הפיקוד (אולם ניתן להכריז על כל מסוף כמסוף פיקוד, בתנאי שאין מסוף אחר בסטטוס זה).

תוכניות כמו Report, Class, Student, Time, Date, מטפלות כל אחת בנושא המוגדר על-ידי שמה. התוכנית Class, למשל, מאפשרת בדיקת עבודתה של כתה שלמה, כאשר בדיסק ניתן לאחסן קובץ של 64 כתות.

התוכנית המרכזית, היוזמת ומנהלת שעורו של תלמיד בודד, היא התוכנית Drill. על-מנת להפעילה, לוחץ התלמיד על מקש הרווח. בתגובה - מבקש המסוף את מספר התלמיד.
לאחר קבלתו קוראת התוכנית את רשומת התלמיד מן הדיסק, מפעילה מונה לתרגילים ומונה לזמן, ומציגה בפני התלמיד סדרה של תרגילים.

מנינו כאן בפירוט יותר את התוכניות המהוות רמה 1. תוכניות אלו מהוות, למעשה, את ה- operating system של מערכת תוא"ם. מערכת הפעלה זו נכתבה במיוחד לצורך נושא תוא"ם והיא שונה בתכלית ממערכת-ההפעלה המקובלת של PDP 11/10. מרגע שמערכת תוא"ם נטענת לזכרון, היא מחליפה את מערכת RT 11/ לחלוטין (RT 11/ היא המערכת בעזרתה נכתבה התוכנית לתוא"ם). הנחיצות במערכת הפעלה שונה, ובעיקר ב- device handlers עצמאיים, נובעת מן הצורך לשלוט במנגנון ה- interrupt ובמיוחד ב- hardware interrupts, וכן בשל החסכון במקום בזכרון הנובע מהחזקת אותן שגרות בלבד החיוניות למערכת הנוכחית.

ברמה 2 כלולות תוכניות שהן לרוב מפעילות (drivers) של התקנים (כגון: מפעיל לדיסק - disk driver , מפעיל למדפסת, מתרגם מחרוזת (String intepreter), קורא/כותב שורה במסוף (Terminal line read/write), ותוכניות אחרות שעיבודיהן מועטים אולם דורשות קלט/פלט רב)

ואילו תוכניות רמה 3 - תוכניות הפיקוח - הן אותן התוכניות שמפסיקות, בכניסתן לפעולה, את פעולתן של יתר הפונקציות עד יציאתן.
תוכניות אלו משנות קדימות או עדיפות של תוכניות אחרות (modify priority), מציגות תוים על המצג, או קולטות תו ממנו, (Input/Output character to/from terminal), מעכבות משימה לזמן מוקצב .(Delay) או מפסיקות אותה (Suspend), מחזירות משימה מופסקת לפעולה (Wake) או מעכבות אותה עד אשר מתפנה ההתקן הדרוש (Delay for disk and printer).

מערכת תוא"ם מאפשרת, על כן, בעזרת שילוב כל התוכניות שנימנו עד כה, לנהל דו-שיח מושלם עם כל תלמיד על יד המסוף שלו, תוך הכונתו, קידומו ובדיקת הישגיו לאורך כל זמן השעור המוקצב לו.
יחד עם זאת מאפשרת המערכת, תודות לתזמון (Scheduling) מתאים, וכמו-כן הודות לעובדה שכל התוכניות נכתבו ב- Assembly , זמן תגובה לתלמיד שהוא קטן מ-2 שניות. כתיבת התוכניות בשפה אחרת, או מערכת הפעלה רגילה, אינן מאפשרות זאת, וזמן תגובה ארוך משתי שניות, גורם להפרת ריכוזו של התלמיד בשעור.

בנוסף על כך מקיימת המערכת קבצים חיצוניים, המאוחסנים על תקליטונים (Rx11). כל תקליטון מכיל כ-2000 קטעים בעלי 64 מילים כל אחד. קבצים אלה שומרים על תוכניות המערכת, על הבעיות המילוליות, על מאגר סטטיסטי (אוגר את מנת הכשלון עבור כל תרגיל לאורך זמן). כמו-כן נשמרים ומעודכנים קבצי התלמידים, קבצי הכתות וקבצים אופציונליים לשמירת נתונים סטטיסטיים נפרדים לכל בית-ספר.

<u>בסיכום:</u> מערכת תוא"ם מהווה הגשמה, בציוד צנוע, של רעיון CAI בדוק ומנוסה. היתרונות שמעניק השימוש בציוד מודרני סטנדרטי הם בעיקר:

1. תכונותיו העדיפות של המצג (CRT) על פני ה- TTY הותיק ממנו. תכונות אלו הן:

א. <u>המהירות</u> הגדולה יותר של הופעת הודעה מודפסת.

ב. היכולת לקבוע <u>מצביע</u> אופטי.

ג. היכולת <u>להבזיק הודעות</u> במקומות שונים של התוכנית.

ד. האפשרות <u>למחוק תוים</u> באופן שהם יעלמו באמת.

ה. היכולת להציג <u>נוסחאות ושברים</u> בצורה המובנת לתלמיד.

כל תרגיל מהווה תהליך המורכב משני שלבים: בשלב הראשון מיוצרת מחרוזת תוים המכילה את התוים (לרוב: מספרים), המוצגים לפני התלמיד, וגם את התוים המרכיבים את התשובה הנצפית מן התלמיד.

לדוגמה, מחרוזת אפשרית היא:

25 + 37 = 62

בה התוים המודגשים בקו מתחתם הם התשובה המצופה.

התוכנית מציגה תחילה, בסקירה ראשונה, את תוי הבעיה ומדלגת על תוי-התשובה. בסקירה שניה היא מציבה מצביע על כל אחד ממקומות תוי התשובה ומצפה לתשובה הנכונה מצד התלמיד, ובזמן המוקצב לו. אם הדפיס תו לא נכון, תופיע מיד על המסך הודעת שגיאה והמצביע יחזור לתו החסר. לאחר 3 הזדמנויות לספק את התשובה הנכונה, ניתן דווח על תשובה מוצלחת או על כשלון, (במקרה אחרון זה מוצגת לתלמיד התשובה הנכונה לפני היציאה). Drill מעדכנת את המונים המתאימים, בודקת אם יש להעלות או להוריד את רמת התרגילים בהמשך ואז, אם עדיין לא חלף הזמן המוקצב לשעור מלא, שהוא 10 דקות, חוזרת למחזור נוסף של יצירת שאלות והצגתן.

במקרה של מסלול הבעיות המלוליות, המצב הוא שונה במקצת. כאן החלטנו לתת לתלמיד את האפשרות להשתמש במחשב כמכונת חישוב לשם קבלת תוצאות ביניים.

אם לדוגמה, הבעיה במילים מוליכה לתשובה שהיא תוצאת החישוב:

35 + 27 x 19

יכול התלמיד להתחיל את פתרונו על ידי הדפסות:

27 x 19 =

ולכך יגיב המחשב בהדפסת: 513.

התלמיד ימשיך וידפיס: 35 + (או, לחלופין = 35 + 513)
ולכך יגיב המחשב ע"י הדפסת 548, ומספר זה יוכר כתשובה נכונה.
יש לשים לב שאילו הדפיס התלמיד = 27 + 35 כצעדו הראשון, יש בידינו לעוצרו מיד, כי בדרך זו לא יגיע לעולם לתשובה המתאימה.
מצד שני, ייתכן שצעדו הראשון יהיה: = 27 x 19, והתוכנית מצפה גם לתשובה זו.

קיימות עוד כמה הוראות שרות כגון: Exercise - המאפשרות לפיקוד לבדוק תרגיל בודד, ומשמשת לצרכי debugging - מציאת שגיאות המערכת, או Copy Disk המאפשרת החזקת תקליטי גיבוי (Backup disks). תוכניות נוספות מטפלות בניקויו של תקליט או של קובץ מסוים בתקליט, בעדכון רשימות התרגילים המילוליים, בחישוב אחוז הכשלונות בכל אחד מן התרגילים, ועוד.

במאמר מוסגר נציין שהבעיות המילוליות, המהוות מסלול לימוד מיוחד, מרכיבות כיום כ-70 רמות, ובכל רמה יש כ-10 בעיות כאלה.

2. צמצום עלות הרכישה והאחזקה וכן צמצום ה- Off-time של המערכת.

3. היכולת של הציוד לשמש בו-זמנית בהחזקת מערכות נוספות, אם בשדה-החינוך, ואם בשטח המינהל.

4. הביצוע המעולה של הציוד המודרני, הן בקיבול, והן בזמן התגובה. לדוגמה - המערכת הבנויה היום על 16 מסופים, תוכל לתמוך ב-32 מסופים, ללא ירידה ברמת הביצוע.

המודולריות של המערכת, עם עין צופיה לפיתוחים אפשריים נוספים, ובכוונים שונים רבים, מאפשרת תוספות והרחבות: הוספת תוכניות לפיתוח הנושא הנוכחי, תוספת מסלולי לימוד ושינויים במסלולים הקיימים, והוספת תת-מערכות CAI, כגון: תרגול ואמון בקריאה או בלימוד שפה זרה.



בימים אלה עוברת המערכת הרצה, ולצורך זה מבקרים בה בלווי מוריהם, ילדים המבלים בקייטנה ומשלבים שעורי חשבון לתוכנית בלוייהם האחרים.
כדאי אולי לעמוד קצת על האופן בו מקבלים הילדים את הנסוי. ראשית יש לציין כי הפגישה עם המכונה לא מביכה את התלמידים הצעירים וההשתלטות על תפעול לוח המקשים היא מהירה.

נראה גם ששעת התרגיל מתקבלת בחדווה ולא מתוך אונס. זאת יש גם לבחון לאור העובדה שהניסוי הנוכחי נערך על ילדי קייטנה, ואלה מגיעים אל המסופים היישר מן הבריכה. מתוך מראה פניהם והתנהגותם ניתן להסיק שיש ב-10 דקות התירגול האלה משום משיכה אמיתית. בתחילת התירגול, אפשר לראות כיצד מגיעים הילדים בריצה אל מקומותיהם; במשך זמן הישיבה ליד המסופים הם נראים מרוכזים במסך שלפניהם ואינם משיחים ביניהם; בתום הישיבה יש אחדים המשתוקקים לשבת ולבצע תירגול של 10 דקות נוספות פעם אחת ואפילו מספר פעמים. השאר מתקבצים יחד ומחליפים חוויות ומשווים הישגים.

נראה, אם כן, שהניסוי מוצלח גם מן הבחינה של קניית לב התלמידים, דבר המוסיף ללא ספק להערכת הצלחתו מבחינה חינוכית-לימודית.

שיטת הוראה בנושא הכרת המחשב מערכת הפעלה

רוט מאיר - המכללה הטכנולוגית, אורט, גבעתיים.

רקע

ניתן לומר היום, שכבר נקבעו דפוסים מתודיים בהוראת שפות תיכנות. דפוסים אלו השתכללו והגיעו לרמה נאותה ודי אם נזכיר את שפע החומר הקיים בתרשימי זרימה ובמיגוון הספרות העוסקת בהקניית שפות תיכנות. יש לציין ששפע זה עומד בניגוד לעוני שקיים בהקניית לימודי הכרת המחשב: מערכות הפעלה, דרך פעולתו של המחשב בביצוע תוכניות, עולם המושגים של מרכיבי התוכנה. כאן המקום להעיר שאותם הספרים המאויירים הקיימים בנושא, רובם פשטניים ומסתפקים בהסברים השווים לכל נפש - מעין השכלה כללית בנושא.

נבחן תחילה את הקשיים המתעוררים בדרך הוראת הנושא:

א. בלמדנו שפת תיכנות כלשהי אנו מצויידים בכלי תירגול מאין כמוהו - הרצת תוכניות במחשב בשפה הנלמדת. תרגול זה לא רק משלים את התמונה של שעורי התאוריה אלא אף גורם להעמקה והגברת ההנעה של התלמיד במקצוע. אם נרצה לבצע תרגולים מעשיים להבנת המערכת נזדקק למשאבים מרובים אשר לא תמיד הם בנמצא.

ב. ידיעת שפת הסף של המחשב שעל מערכתו אנו רוצים ללמוד, איננה מן המשימות הקלות, לעומת זאת הדרך להבנת המערכת מובילה אותנו לעיתים קרובות לשפות אלו.

ג. מקובל לחשוב הרוב החומר הינו הרצאתי-סיפורי, כלומר, אינו תירגולי. רוב הדברים ניתנים להצגה של מסירת מידע והו לא. ברמות מסוימות מתקבל הרושם שזה הצד ה"הומניסטי" של לימודי המחשב. עובדה זו רק מגבירה את הניתוק מהמחשב, בעוד שהמחשב הינו עיקר עסוקנו.

המתודה

לפני הגדרת השיטה חובה עלינו לשאול: למי מכוונים דברנו?
יש להבהיר נקודה זו לפני שנפליג לכלליות רבה מדי. התשובה לשאלה זו: לתלמידי תיכון הלומדים במגמה למחשבים (אם במגמה טכנית, אם במגמה ריאלית עיונית) וכן ניתן לתלמידים ברמה על-תיכונית בשלבי לימודיהם הראשוניים.

ההנחה היא שהתלמיד בקי לפחות באחת השפות העיליות הנפוצות. בדוגמאות עליהן נרחיב את הדיבור, נעשו כל התוכניות בשפת - BASIC שפה זו נבחרה לאחרונה לשפת הלימוד בבתי הספר התיכוניים - אם כי כל אחת מהשפות האחרות טובה באותה מידה לצורך המתודה.

השיטה מנסה לתת התשובות לבעיות שהועלו וזאת בעזרת תרגילים פשוטים יחסית ושאינם מצריכים יותר מאשר ביצוע תרגיל תיכנותי באחת השפות העיליות. התרגילים הם תוכניות סימולציה, אשר אגב כך יכולים ללמוד גם על עקרונות הסימולציה, אולם בעיקרו של דבר הם נועדו לטפל במושגים תיאורטיים על ידי מפגש קונקרטי, מעשי, עמהם בתוכניות. להבהרת הנושא נדגים את השיטה על ידי שתי דוגמאות.

סימולציה למערכת עבודה במקביל (מערכת MULTY-PROGRAMING)

רוב המחשבים עובדים במערכת עבודה במקביל ונושא זה הינו מן הבסיסיים בהבנת המערכת. כדי לעמוד על חלוקת משאבי המחשב בין התוכניות היושבות בזיכרון ולעקוב מקרוב אחר הבעיות המתעוררות בגלל שיתוף של המשאבים בין התוכניות השונות - ניתן לעשות את התוכנית שהוראותיה יפורטו בהמשך.

הנחיות לתרגיל

1. בזיכרון 4 מחיצות היכולות להכיל 4 תוכניות (שלב זה ניתן להכללה למספר שונה של מחיצות וכן למחיצות בעלות גדלים שונים אשר יכולים להכניס שיקולים נוספים על אלו המוזכרים בהמשך).
2. במצב התחלתי כל המחיצות תפוסות על ידי תוכניות של משתמשים.
3. את משאבי המחשב (נסתפק לצורך פישוט הבעיה בשני משאבים חשובים: יח' עיבוד מרכזית, ה- CPU ומערכי קלט פלט, ה- I/O) מקבלות התוכניות לפי סדר ישיבתן בזיכרון: תוכנית ראשונה

ראשונה בקבלת ה- CPU או ה- I/O, בהנחה שהתוכנית עסוקה רק באחד משני הדברים, ולא עסוקה בשניהם בו-זמנית.

4. כאשר תוכנית מקבלת את ה- CPU לשימושה, היא מקבלת אותו לשתי יח' זמן.

5. כאשר תוכנית מקבלת את ה- I/O לשימושה, היא מקבלת אותו לשלוש יח' זמן.

6. הדרישה למשאבי המחשב נקבעת בתחילה. לכל תוכנית היושבת בזכרון מוצאים ארבעה מספרים אקראיים שלמים (תחום המספרים יכול להיות מוגדר באופן שרירותי וניתן לשינוי כדי לבדוק הבטים שונים של אותה בעיה), כאשר המספר האקראי הראשון הינו עבור זמן CPU, המספר השני עבור זמן I/O השלישי עבור CPU ואילו הרביעי עבור I/O (גם זה ניתן לשינוי, הן במספר ההופעות של משאב זה או אחר והן בסדר ההופעות גם כאן כמקודם זה יכול לאפשר הבטים נוספים).

7. חלוקת המשאבים היא בהתאם לצורך ובהתאם ליחידות הזמן שנקבעו בסעיפים 4 ו-5.

8. כאשר תוכנית גומרת את ביצועה ניתן להכניס במקומה תוכנית חדשה ובכן להציג עבודה דרישות למשאבי מחשב.

הדו"חות שאפשר להפיק מעבודה זו

1. הדפסה המפרטת לכל תוכנית את ההסטוריה שלה: מועד כניסתה לזכרון, משך זמן ביצועה, משך זמן שהייתה בזיכרון ומועד יציאתה מהזכרון.
2. תאור מצב רגעי בזכרון: תמונה כוללת על מצב המשאבים והמחיצות בכל יחידת זמן.
3. ד"ח סטטיסטי על אחוז נצילות של ה- CPU וה- I/O; על ידי הרחבה פשוטה של העבודה יש מקום לבדיקת אחוז הנצילות של הזכרון.

המושגים הניתנים לבאור ולהבהרה על ידי התוכנית

1. המחשה של מצב רגעי במערכת עבודה במקביל.
2. המושג - ELAPSED-TIME, זמן שהיה כללי בזכרון.
3. המושג - WAIT והסיבות שיכולות לגרום להיווצרותו.
4. נצילות או חוסר נצילות של משאבי המחשב ברגע מסוים, שהרי יתכן מצב בו לא תהיה דרישה לאחד המשאבים וזאת במקרה שכל התוכניות זקוקות למשאב האחר.

5. המושגים CPU BOUNDED וכן I/O BOUNDED מתייחסים להיות מוגדרים על ידי דרישה גדולה של CPU בתוכנית מסוימת לעומת דרישה גדולה של I/O בתוכנית אחרת. כך ניתן לראות את השילוב בין שני הסוגים במסגרת חלוקת המשאבים.

6. על ידי חלוקת משאבי המחשב שלא בהתאם לסדר ישיבתם בזיכרון, ניתן ליצור PRIORITY, סדר קדימויות של תוכנית אחת ביחס לשניה.

<u>דוגמה</u>

הדו"ח המובא להלן הינו עבודתו של תלמיד כיתה י' באורט גבעתיים. בהנחה שהדרישה ההתחלתית למשאבים עבור ארבע התוכניות הראשונות היושבות בזכרון הינה לפי הטבלה שבסמוך. המספרים שבכל שורה מתארים את משכי הזמן שהתוכנית זקוקה למשאב המצויין בראש העמודה. הדרישה למשאבים היא לפי הסדר של העמודות בשורה.

תוכנית	CPU	I/O	CPU	I/O
A	2	3	4	5
B	2	1	2	2
C	4	3	3	5
D	4	4	4	2

נקבל דו"ח על 14 יח' הזמן הראשונות להפעלת המערכת

TIME	A	B	C	D
1	CPU	WAIT	WAIT	WAIT
2	CPU	WAIT	WAIT	WAIT
3	I/O	CPU	WAIT	WAIT
4	I/O	CPU	WAIT	WAIT
5	I/O	WAIT	CPU	WAIT
6	WAIT	I/O	CPU	WAIT
7	WAIT	WAIT	WAIT	CPU
8	WAIT	WAIT	WAIT	CPU
9	CPU	WAIT	WAIT	WAIT
10	CPU	WAIT	WAIT	WAIT
11	WAIT	CPU	WAIT	WAIT
12	WAIT	CPU	WAIT	WAIT
13	WAIT	I/O	CPU	WAIT
14	WAIT	I/O	CPU	WAIT

PROGRAM IN PARTITION 2 HAS FINISHED
WAS IN THE MEMORY FOR 14 UNITS OF TIME.
% TIME WAS WAITING 50

סימולטור לביצוע אלגוריתמים (בקיצור סל"א)

בדרך כלל כשמנסים לחדור מעבר לשפות עיליות, הנלמדות בשלבים הראשונים של הלימוד, מעמידים את הלומד בניתוק מדרך ביצועה של התוכנית שלו עצמו. אחד המחשבים הקדים הגורמים לניתוק הוא, כפי שכבר הזכרנו במבוא, לימוד שפת הסף. הבעיה העיקרית עם שפות הסף שלימודן מצריך מאמץ מרובה ודרך הרצתם מורכבת יותר, לכן נוצר מצב לא בריא שמצד אחד למד התלמיד שפות עיליות ומאידך למד את מרכיבי המערכה על דרך הרצאתית-סיפורית, שיטה אשר איננה ממחישה ואיננה מטמיעה את החומר אלא משאירה את התלמיד במעין "ריחוף קל".

קיימת גם בעיה נוספת: לימוד שפת ASSEMBLER של מחשב מסוים איננו הופך את הלומד למבין בשפות ASSEMBLER של מערכות אחרות, ישנם הבדלים ניכרים בין החברות השונות ובעיקר בין מחשבים גדולים למיני-מחשבים. אמנם הטענה המקובלת שלימוד שפת סף מטיפוס מסוים יש בו כדי להקנות דגם ומושגי יסוד בהכרת המחשב. אך אם זה הנימוק, הרי, לדעתי, ניתן לעשות זאת בפחות מאמץ וזאת אם לא נתיימר ללמד שפת סף אמיתית אלא נלמד שפה שתהא פונקציונאלית לצרכינו ואשר תעסוק בכל אותם המושגים שברצוננו להקנות.

אך אם נישאר עיקשים בדעתנו ללמד שפת ASSEMBLER ניתן להשתמש בתרגיל זה כמבוא וכהכנה.

מרכיבי שפת סל"א

בשפת סל"א ניתן להבחין בשתי שכבות: כאשר השכבה הראשונה כוללת 4 פקודות יסודיות וקבועים עשרוניים שלמים, השכבה השניה כוללת את הפקודות והקבועים של השכבה הראשונה ועליה נוספו עוד 9 פקודות וכן קבועים עשרוניים בנקודה צפה. במאמר זה נסתפק במסגרת הצרה של השכבה הראשונה, שכבה אשר מסוגלת לעמוד בפני עצמה ולצורך הבנה עקרונית של הסימולטור ניתן להסתפק בה. מבנה הפקודות דומה בעיקרו לפקודות מכונה והן כוללות שלושה מרכיבים עיקריים;

מבנה פקודת סל"א

ADDRESS כתובת	OP CODE קוד הפקודה	OPERAND נתון

קוד הפקודה הינו סימבולי ורומז על תפקידו, עובדה זאת מבדילה אותו בעיקר משפת מכונה והופכת אותה לשפת - ASSEMBLER

לדוגמה

נתון	קוד פקודה	כתובת
012	LDA	003

זאת פקודה אשר כתובתה היא 3 ומהותה LOAD ACCUMULATOR פירושו של דבר לטעון את הצובר בתוכן של הכתובת המצויינת בנתון. כפי שניתן לראות לפנינו פקודה אופיינית לרמה זו של שפת סף. צורת הצגתה של פקודה זו מבחינת גודלה בהתייחס לבית ולמילה, חורג מתחום מאמר שמציג את העקרונות בלבד.

סוגי הפקודות של השכבה הראשונית

1. אחסון ושליפה של נתונים לתוך היחידה האריטמתית וממנה.
2. ארבע פעולות חשבון היסודיות: חיבור, חיסור, כפל וחילוק.
3. הסתעפויות: דילוג בלתי מותנה, דילוגים מותנים בהתאם לתוכן של היח' האריטמתית.
4. פעולות קלט פלט.
5. פעולת עצירה של מהלך התוכנים.

הקבועים, כאמור, שלמים בלבד ומקומם בסוף התוכנית.

כמו כל שפת תיכנות כך גם שפת סל"א מיועדת להרצה על ידי מחשב אלא שבניגוד לשפות סימבוליות אחרות המבוצעות על ידי תירגומן ישר לשפה מכונה, הרי שפת סל"א עוברת עיבוד של סימולטור הכתוב בשפה עילית כלשהי ואשר מקבל את התוכנית בשפת סל"א בקלט, תוצאת העיבוד של שפה זו יהווה עבור התוכניה את הפלט. מאליו מובן שתוכניה זו צריכה לטפל בבעיות תחביריות ובבעיות הקשורות בביצוע התוכניה במקובל לגבי תוכניות בכלל. אך למרות כל האמור אין כותב התוכנית בשפת סל"א חייב להתחשב בעובדה שישנו גורם מתווך בינו לבין המחשב.

בהמשך ננסה לעמוד בקצרה על עקרונותיו של הסימולטור וכן להראות דוגמה של תוכנית המורצת דרך הסימולטור.

הסימולטור של סל"א

מהדר - COMPILER

כמו כל מהדר כך גם כאן יש לו שני תפקידים עיקריים:

1. תרגום התוכנית משפה סימבולית לשפת מספרים (שפת מכונה).
2. איתור שגיאות תחביריות.

למרות ששני התפקידים הללו שונים במהותם הרי הם משולבים זה בזה בדרך ביצועם, כי קל לאתר שגיאה תחבירית אגב נסיון תרגום הפקודה. אם הפקודה נכתבה כהוגן מבחינה תחבירית אז המהדר רושם את הקוד בשפת מכונה (קוד זה ישמש לקשר את הפקודה עם הסברוטינה המבצעת אותה).

בשגיאות התחביריות ניתן למצוא שני סוגים עיקריים:

1. פקודה לא חוקית : פקודות שאינן קיימות, או שנרשמו בפורמט לא נכון.
2. סדר לא נכון של פקודות : סידור קבועים בתוך סדרת הוראות, העדר הוראת סיום.

בתום שלב זה - אם אין תקלות - עוברים לשלב הבא.

מבצע - PROCESSOR

החלק המבצע בסימולטור מקבל תוכנית מתורגמת בקלט. התוכנית נסרקת פקודה אחר פקודה לפי סדר הופעתה, אלא אם כן היא מכילה הוראות דילוג. הביצוע של ההוראות עצמן נעשה על ידי הפניה לסברוטינות מתאימות לכל פקודה ופקודה, מקום שם מתבצעת הפקודה, רק בשלב זה באה לידי ביטוי הוראת קלט על ידי בקשה לנתון חיצוני. גם הוראת הפלט נעשית על ידי פניה לאמצעי פלט ושליחת נתון להדפסה.

הטיפול בשגיאות הינו אחד מתפקידי המבצע ולכן נסקור את השגיאות ואת דרך הטיפול בהן.

דפיי ביצוע ודרך טיפולם

1. חיפוש או בניה לכתובת שמחוץ לתחום - ADDRESSING
2. שימוש לא חוקי ביח' האריטמטית.

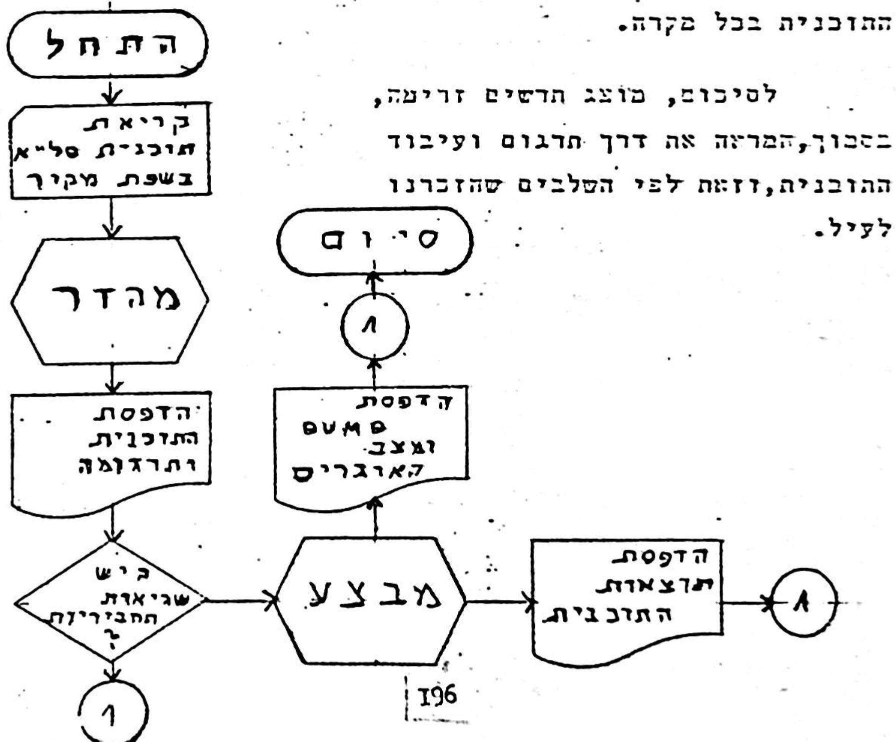
בשני המקרים הללו מתבצעות הפעולות הבאות:
מודפס תוכן הזכרון בזמן התארעות התקלה - DUMP . במקרה זה מקבל המתכנת את מצבם האחרון: של התוכנית ושל הקבועים או שטחי עבודה הקשורים לתוכנית.

3. לולאה אינסופית של התוכנית עצמה.

על בעיה זו לא ניתן להתגבר אלא על ידי הקצאת זמן למשך ביצוע התוכנית. לסימולטור ישנו TIMER אשר מתקדם בקצב הביצוע של הפקודות, כאשר כל פקודה גורמת להתקדמות ה-~~TIMER~~ ביחידות זמן מסויימות בהתאם לאופי הפקודה. כאשר עוברת התוכנית את "הזמן המוקצב" - יפסק ביצועה... כמו כן ניתן להשתמש ב-TIMER כמודד את משכי הביצוע של התוכניות בכלל.

לפני שהתוכנית נכנסת לסימולטור, יש לספק לה מידע: על מספר הפקודות, מספר הקבועים וכן יש לספק מידע על זמן ביצועה המשוער וזאת כדי שהסימולטור ידע מתי יש להפסיק את ביצוע התוכנית בכל מקרה.

לסיכום, מוצג תרשים זרימה, בסבוך, המראה את דרך תרגום ועיבוד התוכנית, וזאת לפי השלבים שהזכרנו לעיל.



דמות העיסדק בסל"א

1. המורה המלמד מכין את הסימולטור באחת השפות העיליות העומדות לרשותו, מאחסן אותו על אמצעי אחסון (רצוי דיסק). התלמיד המשתמש בסימולטור מוסר לו את התוכניות שלו בשפת סל"א ומקבל את התוצאות לאחר שהתוכנית עובדה על ידי הסימולטור.
2. התלמיד מכין את הסימולטור בעצמו ומנסה אותו על ידי תוכניות בשפת סל"א. אין ספק שבדרך זו מתרגל התלמיד פרק בסימולציה ועומד בפני בעיה תיכנותית יפה.

דוגמה

הדוגמה המובאת בזה היא הרצה על סימולטור שבנה תלמיד שנה א' בבית הספר להנדסאים באורט גבעתיים. הסימולטור נעשה ב* Basic

```
IS YOUR PROGRAM WAS WRITTEN IN SELA? YES
HOW LONG IS THE PROGRAM ? 9
HOW LONG IS ONLY THE ORDERS ? 6
HOW MUCH TIME IT WILL TAKE (COMPILATION)? 10
WRITE YOUR PROGRAM AFTER EVREY QUESTION MARKWRITE ONLY ONE LINE
 1 ? LAC007
 2 ? DVI008
 3 ? MPY008
 4 ? STO009
 5 ? WWD009
 6 ? HLT000
 7 ? 001000
 8 ? 000007
 9 ? 000000
THE ORDERS
LABEL          *             AFTER          *          BEFORE
 1             *              210  7        *          LAC007
 2             *              270  8        *          DVI008
 3             *              260  8        *          MPY008
 4             *              220  9        *          STO009
 5             *              330  9        *          WWD009
 6             *              340  0        *          HLT000
WILL YOU CORERCT ANY ORDERS? NO
O. K.
THE CONSTANTS
LABEL          *             AFTER          *          BEFORE
 7             *              1000          *          001000
 8             *              7             *          000007
 9             *              0             *          000000
THE COMPILATION TOOK 1.5 MSC.
TO EXECUT IT(OR COMPILATION IS ENOUGH)? YES
O K.
EXECTION
HOW MUCH TIME IT WILL TAKE(EXECTION)? 45
 994
EXECTION TOOK 4.3  MSC.
```

כפי שניתן לראות אף מההצגה הקצרה של סל"א, המושגים בהם משתמש הסימולטור הם רבים, דוגמה שעל ידי בנייתו ניתן להקיף את חלק הארי של מושגי מערכת ההפעלה הבסיסיים, ועל כל זה יש להוסיף שהתרגיל עצמו מהווה לקח מאלף לגבי סימולציה בכלל.

<u>עבודה בכיוון מעשי</u>

העבודה נמצאת בשלבי ניסוי, נזכיר רק את הנסיונות שנערך לגבי שתי הדוגמאות המובאות במאמר. הנסיונות נעשו במכללה הטכנולוגית באורט גבעתיים ובבית הספר התיכון שלידו.

הסימולציה למערכת עבודה במקביל נוסתה בכיתה י' במסגרת מבוא למחשבים במגמה למחשבים.

הסימולטור של סל"א נוסה בכיתות י', י"א במסגרת מבוא למחשבים ומבוא למערכות הפעלה במגמה למחשבים. כך נוסה במכללה בשנה הראשונה של הנדסאי מחשבים במסגרת מבוא לשפת ASSEMBLER.

ביבליוגרפיה

1. רוט מאיר/חוברת סל"א/פנימי, אורט גבעתיים.
2. פורסייט, קינן, שטרנברג/יסודות מדעי המחשב/המכון לאמצעי הוראה.
3. צבי אריאל/מערכת 370/י.ב.מ. ישראל.
4. FORSYTHE, ORGANICK, PLUMMER / COMPUTER SCIENCE – PROJECTS AND STUDY PROBLEMS / JOHN WILEY & SONS 1973

ON THE COMPARATIVE EVALUATION OF TEACHING SYSTEMS UNDER COMPETITIVE CONDITIONS.

Dr. Luis Osin
Centre for Educational Technology
Ramat Aviv, ISRAEL

INTRODUCTION

Although CAI developers are mostly fascinated by the intellectual aspects and open possibilities of the subject, they feel the need to prove to others, specially administrators, the advantages that CAI will bring to education. Of course, this is not unique to CAI; every technological innovation trying to find its place in the educational system has to prove its right to exist. Not surprising, when we remember that education is usually the biggest national enterprise. A wealth of comparative experiments were thus conducted, a survey of which was recently published (Jamison, Suppes, and Wells, 1974). In the experiments related to CAI the emphasis for the moment is not on cost but rather on whether traditional instruction may be substituted totally or partially without detriment or, hopefully, with benefit to the students. This paper will provide a critical view and some suggestions on comparative evaluation.

ORGANIZATION OF EXPERIMENTS

After an initial stage of research experiments, actually feasibility studies for hardware and software testing, the trend now is to compare systems under real conditions. We will refer, in this paper, to experiments conducted at university level, taking advantage of this author's personal experience at two universities in the United States, during the academic years 1974 - 1975 and 1975 - 1976. It is important to realize that in very few places (even in the United States) it is possible to conduct large scale experiments which require extensive CAI facilities.

The typical organization for comparative evaluation is to teach two or more groups under different conditions and to compare results. The program to be taught must be the same for all groups, as well as the achievement tests for the students. When comparing different methods all additional effects must be equalized or filtered. For instance, student populations and environment must be equivalent for all groups.

The elements which are considered in order to judge the superiority of one system or condition over another are:

1. Students' grades or test scores.
2. Number of contact hours between students and teacher or system.
3. Cost.
4. Evaluation questionnaires filled by the students.

The three first items provide objective measures, while the fourth introduces a subjective component.

After this introduction, the body of this paper will try to show that:

1. Those objective components may be insufficient for correct decision making.
2. An important objective component is missing in the list, and has not been measured in the experiments presented in the literature.
3. The subjective comments cannot be taken at face value.
4. In addition to the Hawthorne's effect it is necessary to consider what will be called "the survival principle", in order to account for certain experimental results.

Two large scale experiments will be presented, both conducted at the Urbana campus of the University of Illinois, by Professor Richard Montanelli, Jr. A smaller scale experiment conducted by this author at the University of Michigan will be analyzed also.

The experiments at the University of Illinois refer all to the same course: CS105, which is a required course for business majors, an introduction to computational techniques where students learn FORTRAN programming. The traditional way of teaching this course is with two weekly hours of lecture (large groups) and one hour of discussion (small groups). The students must write and run FORTRAN programs (machine problems), and consultants are available to help them. Final grades are based on machine problems, two hourly exams, and a final exam.

TEACHER DIFFERENCES

In the first experiment to be presented the only difference in treatment consists in having different lecturers for different groups. This first analysis serves particularly well the purpose of this paper because there are no extraneous effects introduced by a new technology like CAI.

Four lect[illegible]s taught six groups, with an initial total population of 868 stud[illegible] At the end of the course the students filled evaluation questionnaires regarding lecturers and course. From the first 16 items, four were selected ("a priori") for this analysis as having special relevance in the characterization of the lecturers. They are:

1) Presentation of course material.
2) Ability to explain difficult material.
3) Ability to create interest in course material.
4) Overall effectiveness as an instructor.

For each one of the items each student had to select one of five options. For item 1, for instance, they were:

a) Exceptionally well organized and clear.
b) Consistently organized and clear.
c) Reasonably organized and clear.
d) Sometimes has difficulty in presentation.
e) Usually unable to convey material well.

Mapping answers into integer values from 0 to 4, where 0 corresponds to the worst case and 4 to the best, it is possible to compute averages and organize the following table for their display:

ITEM	LECTURER			
	A	B	C	D
1	2.49	2.77	1.90	1.42
2	2.93	3.08	2.39	2.21
3	3.07	3.62	2.71	2.62
4	2.73	3.06	2.23	1.87

A two-way analysis of variance for each one of the items shows that the differences are statistically significant at the .001 level. It is very clear that lecturer B is the best, and lecturer D the worst, according to the students. Informal conversations with the staff strongly confirm this evaluation.

The students' average grades, classified by lecturer, are:

Lecturer	Machine problems	1st hourly exam	2nd hourly exam	Final exam
A	142.4	84.1	42.5	144.9
B	147.2	87.7	42.9	147.1
C	134.9	84.4	39.3	138.3
D	151.3	87.2	41.7	146.9

The grades were not assigned by each lecturer individually. Machine problems were graded by assistants, and a unique team was assigned to each exam question, in order to insure uniformity in the grading criteria.

It is very clear that the results obtained by the students corresponding to the "best" and the "worst" lecturers are almost identical. Furthermore, another objective measure (drop-out rate) shows minimum drop-out for the worst lecturer (!).

We will not accept the easy interpretation, which results from the consideration of those objective components, that quality of teaching has no influence on the educational system. The size of the groups precludes another "easy" interpretation, which is that by chance the students assigned to lecturer D were better than the others.

Our thesis, which will be supported by further examples, follows:
"Principle of survival": Students competing for grades within a university course will reach a level of performance correlated to their individual capabilities, irrespective (within limits) of teaching techniques or teacher qualifications.

A student receiving instruction from a poorer educational subsystem will compensate by means of additional personal effort, reflected in a higher number of individual study hours. The number of hours which students study has not been measured in any experiment of this kind, at least to this author's knowledge.

It is clear that a student forced to work on his own (particularly in an area like programming) during the whole course will acquire the necessary training to perform efficiently in the exams. On the other hand, weaknesses of an educational system which students overcome on the basis of an extraordinary time investment have a negative impact on the students performance in other parts of the system.

PARTIAL SUBSTITUTION OF LECTURES BY CAI.

The second experiment we present is fully described in Montanelli (1976). Approximately half of the students in CS105, the course whose traditional instruction was previously presented, were taught with one hour of lecture, one hour of CAI (under the PLATO system) and one hour of discussion.
In order to analyze the influence of the substitution of one hour of lecture by one hour of CAI while, at the same time, filtering effects which could be introduced by differences in lecturers or schedules, the students in the experiment were randomly assigned to four groups under the following scheme:

		PROFESSOR	
		A	B
SCHEDULE	9 - 10	CAI	non-CAI
	10 - 11	non-CAI	CAI

On Mondays, the four groups received one hour of lecture; on Wednesdays only the groups marked non-CAI received a lecture. Students in a CAI group were assigned one weekly hour of terminal time. Each one of the four groups had approximately 100 students.

Students' grades and drop-out percentages, classified by lecturer and treatment, are:

LECTURER	TREATMENT	Machine problems	1st hourly exams	2nd hourly exams	FINAL exams	DROP-OUT %
A	non-CAI	142	91	62	134	4
	CAI	143	92	65	140	15
B	non-CAI	146	91	62	134	14
	CAI	140	91	62	135	25

The grades are essentially the same for all groups. The only significant difference appears in the drop-out rate, where figures corresponding to lecturer B are 10% above those of lecturer A, and CAI figures are 11% above non-CAI.

Comparative evaluation data among the two lecturers was not available.

The "survival principle" appears again very clearly. Irrespective of differences among lecturers and treatments, the students who decided to stay in the course reached the same level in each one of the four groups.

The high drop-out rate in CAI cannot be attributed to a negative student attitude towards CAI. In effect:

1) A PLATO Student Attitude Survey (Siegel, 1974) showed a very positive attitude towards CAI. In particular, 42.2% of the students indicated that the most important advantage PLATO offers is: "better understanding of the material".

2) The evaluation questionnaires filled by the participating students at the end of the course showed that those in the CAI groups strongly recommended this approach vs. the traditional. On the other hand, students in the traditional mode were indifferent.

As one of the two participating lecturers, this author will incorporate his personal knowledge of some additional facts, not usually published.

a) Very few professors at the Department of Computer Science were interested in the production of the CAI course, so that the lessons were usually produced by students as term projects. These lessons were poor in many cases and bad in a few. When this author tested the lessons, the majority contained conceptual errors. Furthermore, a correct answer coul be rejected by the system due to insufficiencies in the answer-recognition part of some lessons. These elements may have influenced the drop-out rate.
Nevertheless, and strongly reinforcing our thesis that student time is of paramount importance, the evaluation questionnaires showed that "the worst feature of PLATO", according to students was:
"The distance to the terminals" (which were at the north edge of campus). This feature was marked more than double its follower ("lack of human contact").

b) The part of the evaluation questionnaires which relates to this author's performance as a lecturer, shows a clear difference between CAI and non-CAI groups. The non-CAI students had a much better opinion than the CAI students. Being all other things equal, we submit the hypothesis that the reduction in contact time, from two weekly hours to one, had an adverse influence in the personal relationship between lecturer and students. This negative component in class "atmosphere" may be shown by analyzing one particular item in the evaluation questionnaires:
"Adequate arrangements were made when the professor was absent". The answers could vary from "strongly agree" to "strongly disagree". A significant difference appears among the two groups with the CAI group having, of course, the poorest opinion on the subject. The claim that, in this case, the item measures only a subjective component is easy to defend. In effect, this author was never absent from his lectures in this experiment.
The point to emphasize here is that an educational system may suffer from the inclusion of a possibly efficient new subsystem if other parts of the system degrade their performance.

TOTAL SUBSTITUTION OF LECTURES BY CAI.

This author taught an introductory programming course (CCS 273) at the University of Michigan. Students who could not assist to the lectures were offered a CAI option. The experience is fully described in Osin (1976), but suffices here to say that there were no drop-outs among the seven CAI students and they passed with an average grade slightly above that of the non-CAI students who passed (31 out of 39 who started the course). Furthermore, it took an average of 15 terminal-hours to complete the CAI lessons, while the traditional course had forty hours of lecture.

This experience shows again the survival principle in action. As usual, individual-study time was not measured. In this case there is an additional difference, which relates to the material presented; the examples and applications in the traditional course were larger and conceptually richer than those in the CAI lessons. Although the tests related only to the FORTRAN language, it is possible to argue that students in the traditional mode had a possibility of cultural enrichment which was not offered to the others.

SUMMARY

1) Large scale experiments are useless, if we lack adequate control and in depth knowledge of the elements involved.

   In particular, future experiments should measure the students' individual-study time.

2) Although, CAI has already shown its feasibility, its use as a replacement for parts of an educational system must carefully weight the changes in existing parts.

   This consideration is not esential when CAI is used as a substitute for or an addition to an educational system.

3) Comparative experiments which found "no significative difference" between treatments, should be analyzed under the assumptions of the "survival principle".

REFERENCES

JAMISON, D., SUPPES, P., AND WELLS, S. "The Effectiveness of Alternative Instructional Media: A Survey", Review of Educational Research, Vol.44, No.1, Winter 1974,

MONTANELLI, R.G. "Evaluation of the use of CAI materials in an introductory computer science course", 1976 AEDS International Convention, Phoenix, Arizona May, 1976.

OSIN, L. "SMITH: How to produce CAI courses without programming", International Journal of Man-Machine Studies, March, 1976, pp. 207-241.

SIEGEL, M.A., "PLATO Student Attitude Survey", PLATO Evaluation Report, May, 1974, Urbana, Illinois.

קבוצת דיון מס' 7:

"שימושי מחשב בתחבורה אוירית יבשתית וימית"

יו"ר: **כ. צימט**

כ. צימט

י. ורול

ד. רוזן

א. בן כוזרי

ה. מטמון

מערכת "כרמל" - להזמנות ושינוע נוסעים ב"אל-על"

כ. צימט

השם "כרמל" - *Computerized Airline Reservations & Message Switching of EL-AL* נבחר בתחרות פנימית מתוך כמה מאות שמות אפשריים שהוצעו ע"י עובדי החברה. ועדת השופטים מצאה את השם המורכב מהאותיות "כ.ר.מ.ל." כמתאים ביותר.

אם נעמיק יותר בפעולת המערכת הממוחשבת ניוכח ששם זה לא מבטא את מלוא כושרה ותכונתה כי המושג "שירות לנוסע" לא מוצא את ביטויו וזה היעוד האמיתי של המערכת כפי שהיא התפתחה בשלבים המתוארים להלן.

השלב הראשון ניתן לתאר כשלב טרום-טיסתי המתמקד בעיקר בהתעניינות באפשרויות הקשר השונות. הנוסע הפוטנציאלי מעוניין לדעת מהן הטיסות האפשריות במועדים וזמנים רצויים אשר יעבר על מבוקשתו.

בשלב השני יזמין הנוסע (או סוכן הנסיעות הפועל בשמו) מקום בטיסה המתאימה לו ביותר כדי להגיע ליעדו. טיסה זו יכולה להיות בלעדית של חברת "אל-על" עם היעד הוא ברשת הטיסות שלה או טיסתו של הנוסע תהיה מורכבת מקטעים בחברת "אל-על" והמשכן בחברות אחרות, לדוגמא: טיסה מתל-אביב לוושינגטון תתבצע מתל-אביב לניו-יורק ב"אל-על" ומניו-יורק לוושינגטון בחברה אמריקנית.

בשלב השלישי יתכן והנוסע ישנה את דעתו הן מסיבות של נוחיות והן מתוך סיבות אחרות ויחליט לשנות את הזמנתו המקורית לטיסה בתאריך אחר, ליעד אחר או כל פרט מהפרטים השונים שהזמין.

בשלב הרביעי תבוצע ההזמנה. הנוסע יתיצב בשדה התעופה ליד דלפקי הנוסעים ופקיד החברה יצטרך לוודא אם הוא אמנם רשום בטיסה, לאשר לו מקום ישיבה במטוס ולדאוג לכל הבקשות המיוחדות שנרשמו קודם לכן כגון: אוכל מיוחד, סידורים מיוחדים לתינוקות וכדי. באותה עת המידע על הנוסע ומזוודותיו ילקח בחשבון בחישובי משקל המטוס למטרות מבצעיות ובטחוניות.

אם ניקח בחשבון שחברות התעופה הגדולות מובילות מיליונים של נוסעים וגם "אל-על" מתקרבת או תעבור השנה את מיליון הנוסעים, נקבל מושג מהיקף הבעיה.

2/...

התפתחות מערכות הזמנות אלקטרונית

חברות התעופה הגדולות ניסו להתמודד עם בעית ההזמנות אחרי שבאמצעים הידניים כבר לא היה אפשרי להבטיח שרות יעיל ומסודר למיליוני הנוסעים באויר. שיטות אלקטרו-מכניות שונות מבוססות על ממסרים ומאוחר יותר על מחשבים מסוגים שונים הוכנסו לשימוש אבל לא הצליחו להתגבר על הבעיות המורכבות.
לפני כעשר שנים בנתה חברת אמריקן איירליינס בשותפות עם חברת י.ב.מ. מערכת ממוחשבת הידועה בשם "SABER" שהתפתחה מאוחר יותר למערכת "PARS" - *(Programmed Airline Reservations System)*. זאת המערכת הראשונה שאיכסנה במחשבים את שמות הנוסעים וכל פרטיהם ואיפשרה לכל סניפי החברה (יותר מאלפיים) להתקשר בעזרת מסופים בו-זמנית אל המחשב המרכזי, לקבל ולהעביר מידע מיידי.

מערכת "PARS" התאימה במיוחד לסוג התעבורה האוירית הפנים ארצית בתוך ארצות הברית. מאוחר יותר הותאמה לתנועה בינלאומית ע"י שותפות של בי.או.אי.סי ו-י.ב.מ. וידועה מאז בשם - "IPARS" - *International Programmed Airline Reservations System*..
גם "אל-על" בחרה במערכת "IPARS" כדי למכן את תהליך ההזמנות ולייעל את מבצעי החברה.

בנוסף למערכת איפראס הבסיסית כפי שהיא תוכננה ע"י חברת י.ב.מ. ו-בי.או.אי.סי. (אשר בנתיים התאחדה עם חברת בי.אי.איי. וידועה מאז כחברת בריטיש איירווייס), בנו כמה מחלוצי המערכה האלקטרונית - *British Airways, Swissair, KLM etc.* תוספות חשובות ביותר, כגון שינוע הנוסעים כחלק אינטגרלי ממערכת ההזמנות האלקטרונית, שינוי ועדכוני לוחות זמנים בזמן ממשי ללא צורך לשתק את מערכת ההזמנות אפילו לשניה אחת ועוד.

הפרוייקט ב"אל-על"

הנהלת חברת "אל-על" החליטה על ביצוע הזמנות אלקטרוניות לאחר מחקר כדאיות מעמיק ביותר בקיץ 1973. בשלהי מלחמת יום הכיפורים הוגדרו דרישות פרוייקט ובמסגרת המרכז לעבוד נתונים הוקם צוות לא-הירככי ובין-אגפי לטפל בהקמת המערכת. לוח זמנים שנקבע בתחילת 1974 קבע את הפעלת המערכת בחודש ינואר 1976.

האנשים נבחרו והתחילו בלימודים ובהשתלמויות. הוחל בתכנון והקמת בנין, הציוד הוזמן ובמקביל הוגדרו והופעלו כל תקשורת הנתונים.

הנהלת החברה, באמצעות ועדת היגוי בראשות משנה למנכ"ל, נתנה גיבוי מלא לפרוייקט, וכך הוצא לפועל מבצע המורכב מיותר מ-600 פעולות ראשיות ברשת "PERT". המערכת הופעלה בדיוק כפי שתוכנן, ב-16 לינואר 1976.

3/...

מערכת "IPARS"

כדי לאפשר את הזמנת מקומות בעזרת המחשב חייבת המערכת לכלול בסיס נתונים עשיר, מגוון ומעודכן תמיד כי המידע הכלול בו חיוני לבצוע כל הפעולות הדרושות.

כל השאילתות המופנות למערכת הצגים המבוזרים בסניפי "אל-על" דרך רשת התקשרות פונות אל בסיס נתונים זה, שולפות מידע ממנו ומעדכנות אותו ללא הרף.

המערכת המפקחת הידועה בשם "ACP" -"Airline Control Program" היא מערכת בלעדית אשר מצטיינת במיוחד בטפול יעיל ומהיר בשאילתות רבות בו-זמנית. מערכת ה-ACP ידועה עד עצם היום הזה כמערכת הטובה והיעילה ביותר מסוגה. היא אשר מנתבת את השאילתות השונות, שולטת על הקבצים, מארגנת את המידע לפי צרכי הפעולות השונות, שומרת על התורים המשתנים כל הזמן, ומבטיחה את הקשר עם מערכות ממוחשבות של חברות אחרות.

קשר זה מתבצע בדיסיפליבה הידועה כ-AIRIMP אשר מבטיחה אחידות מוחלטת בהעברת מידע מאת ואל מערכת הזמנות של חברת תעופה. המערכת המפקחת ACP, בהיותה מערכת הפעלה ייחודית, דורשת מחשב מוקדש לה ואינה יכולה לפעול תחת כל מערכת הפעלה אחרת כגון ה-OS. תכונה זו מכתיבה את תצורת מחשבים בה ימצא מחשב מוקדש למערכת ההזמנות בזמן ממשי (REALTIME) ומחשב שני, זהה לראשון, אשר יבטיח גבוי לעת הצורך ויאפשר ניסויים על מערכת זמן ממשי מדומה.

מערכת הזמנות ייחודית ל"אל-על" דורשת מחשב בגודל של 370/145, ועל כן היה צריך לרכוש מחשב 370/145 נוסף למחשב 370/145 שהיה קיים מכבר בחברה ומספר כונני דיסקים וסרטים כדרוש להפעלה סדירה של מערכת בסדר גודל זה.

בשלב התחלתי זה מרוכזים כל הקבצים והמידע המצטבר על שמונה כונני דיסקים 3330, כל אחד בגודל של 100 מליון בתים. הרוב הגדול של הנתונים רשומים על הדיסקים בשני מקומות שונים והמערכת יודעת בכל עת למצוא את הנתונים הדרושים לה בעזרת אנדקסים מתוחכמים.

במקביל ובו-זמנית נרשמות כל התנועות על גבי סרטים המלווים את כל התהליכים. ובעזרת סרטים אלה אפשר לבנות את כל המערכת מחדש במקרה של נזק חלקי או כולל. כל ההזמנות המתקבלות בחברה עד לשנה מראש, רשומות בדיסקים, לוחות זמנים וקבצי עזר רבים ומעודכנות כל הזמן. מדי יומיים מורדים מהדיסקים כל הפעולות השוטפות והסגורות כדי ליצור מקום לקבלת נתונים חדשים.

4/...

הציוד

שני המחשבים המצויים במתקן "אל-על" ידועים כמחשב הכחול והמחשב האדום, היחידות ההיקפיות מחוברות בתצורה כמעט סימטרית לכל אחד משני המחשבים האלה.
ראוי לציין שתצורת המחשבים מאפשרת מיתוג כל יחידה בנפרד ממחשב אחד למשנהו וזאת לאפשר גיבוי לכל יחידה הן לצרכי אחזקה מונעת והן להתגברות על קלקול כלשהו.

המחשב האדום ויחידותיו מוקדשים כולם למערכת "כרמלי" תחת *ACP* והמחשב הכחול פועל תחת מערכת הפעלה *OS* לבצוע כל צרכי החברה בעבוד נתונים.

למערכת זמן ממשי קדימות מוחלטת על גבי כל עבודה אחרת. קדימות זו מובנת בברור כאשר מתחשבים בעובדה שכל פעולות החברה בנושא הזמנות ושינוע נוסעים תלויות לחלוטין בפעולה התקינה של מערכת "כרמלי".

במקרה והמחשב האדום יוצא מכלל שימוש מכל סיבה שהיא, מופסקות כל העבודות אשר רצות באותה עת במחשב הכחול אשר מחליף מיידית את המחשב האדום וממשיך להפעיל את מערכת "כרמלי".

אספקות

הצורך בפעולה בלתי מופרעת של "כרמלי" 24 שעות ביממה, 7 ימים בשבוע במשך כל ימות השנה הכתיב גם הקמת תשתית מתאימה לאספקות חשמל ומזוג אויר.

המערכת לאבטחת אספקת חשמל ללא הפסק מבוססת על שלוש יחידות *UPS* - *Uninterrupted)* *Power Supply)* של 150 *KVA* כל אחת, הפועלות כדלקמן:

זרם החשמל מתקבל מהרשת הארצית דרך שני שנאים. זרם החילופין נהפך לזרם ישיר ע"י מישרי זרם בכל אחת מהיחידות ה-*UPS*. זרם ישיר זה טוען מצברים המאפשרים הפעלה תקינה של כל המחשבים בעומס המירבי שלהם במשך 15 דקות.

לא תעבור אלא דקה אחת עד שיכנס לפעולה גנרטור דיזל מיוחד שיספק הוא מצידו זרם חילופין במקום ההספקה של הרשת הארצית שהופסקה. הזרם הישר המתקבל מהמצברים הופך לזרם חילופין בעזרת ממירים סטטים (*Converter*) והם שמזינים את המחשבים.

מערכת זו מבודדת את המחשבים והציוד הנלווה מכל ההפרעות באספקת החשמל מרשת החשמל הארצית. הפסקה טוטלית, הפרעות, עליות או נפילות מתח פתאומיות וכל הפרעה אחרת לא משפיעות על מערכת המחשבים.

5/...

בצורה דומה מובטחת אספקת אויר ממוזג השומר על טמפרטורה ולחות אחידים באולם המחשבים אפילו בימי הקיץ הכי לוהטים באזור נמל תעופה "בן-גוריון".

מסופים

בחירת המסופים למערכת דרשה התחשבות באילוצים רבים ואחד החשובים בהם, התאמת המסופים לדרישות המערכת *IPARS*.
אילוצים אלה הגבילו את המבחר ליצרבים מעטים בלבד.

חברת אלתא אשר החלה בייצור צג ברשיון חברת *INCOTERM* בארה"ב, נבחרה כספק. הצגים המיוצרים על ידה עבור "אל-על" יוצרו במיוחד במתכונת הדרושה לחברות התעופה בכלל ולחברת "אל-על" בפרט.

הצגים עמדו יפה במבחן של עבודה יום יומית ולא רק סניפי "אל-על" בארץ צוידו במסופי אלתא (כ-160 במספר בנמל תעופה "בן-גוריון", בירושלים, חיפה ותל-אביב), אלא גם סניף "אל-על" בניו-יורק עובד עם צגים תוצרת הארץ וגם סניפים אחרים בחו"ל יצוידו במסופים אלה.

תקשורת נתונים

רשת התקשורת של "אל-על" משתרעת על פני הארץ בעזרת קוים שכורים מאת דאר ישראל.
הקשר עם כל ארצות העולם מובטח בעזרת רשת חברת *"SITA"* - *(Societe' Internationale - de Telecommunications Aeronautiques)*.
חברת סיטא היא חברה משותפת לכל חברות התעופה להבטחת קשר בינהן ובין עצמן באמצעות מברקים ובהעברת נתוני מחשב.

כל התקשרות של "אל-על" מפוקחת במרכז קשר המצוייד במיטב המכשור אשר בחלקו פותח לצרכי פקוח מרחוק בחלל החיצון. ציוד זה מאפשר לא רק פקוח על הקווים באמצעים קובנציונלים, אלא גם פעולות בקרה ותיקון מרחוק ללא צורך בטכנאים במקום.

פעילות ההזמנות בארץ ממוכנת. כאמור מאז ה-16 בינואר 1976. סניפי החברה בארצות הברית חוברו באוגוסט 1976, והסניפים האחרים יחוברו אחד אחד לפי שיקולים מבצעיים וכלכליים.

הקשר בישראל מובטח על ידי קוי משרד התקשורת בין הערים ירושלים, תל-אביב וחיפה אל מרכז המחשבים בנמל התעופה "בן-גוריון" הקוים כפולים כדי לאפשר גבוי במקרה של הפרעות.

לצרכי הקשר עם חוץ לארץ מועברות התשדורות ללונדון ולרשת של סיטא דרך הכבל התת-ימי אז לחילופין דרך הלויין בעמק האילה.

כל הסניפים בארה"ב קשורים לניו-יורק בטלפון. ניו-יורק משמש כמרכז לטפול בנוסעים מכל צפון אמריקה, ובעזרת הצגים הממוקמים במשרד הראשי בניו-יורק מקיימים קשר עם המחשב בנמל התעופה "בן-גוריון".

באירופה וביבשות אחרות, רוב הערים יהיו קשורות בצגים גם הן, ודרך רשת התקשורת של סיטא יכול כל סניף וסניף ליצור קשר עם המחשב בארץ. המקומות הקטנים יותר מתחברים ל"כרמל" על ידי טלקס כי כמות הנוסעים אשר בטיפולם אינה מצדיקה התקנת צגים.

במקרה של קלקול במערכת סיטא, ניתן לקיים קשר בעזרת מערכת *"DIAL-UP"*, מערכת זו פותחה על ידי אנשי הקשר ב"אל-על" והיא מאפשרת, תוך זמן קצר ביותר, לעלות על קו דואר ולהעביר תקשורת נתונים עד שרשת סיטא שוב חוזרת למצב תקין.

זמן התגובה של המחשב על כל שאילתא לא עולה על שלש עד ארבע שניות ב-90 אחוז של כל המקרים.

חברת "אל-על" הצטרפה גם היא לחברות התעופה הבינלאומיות ומפעילה מערכת זמן ממשי להזמנות ולשנוע נוסעים באמצעות מערכת ממוחשבת.

./...

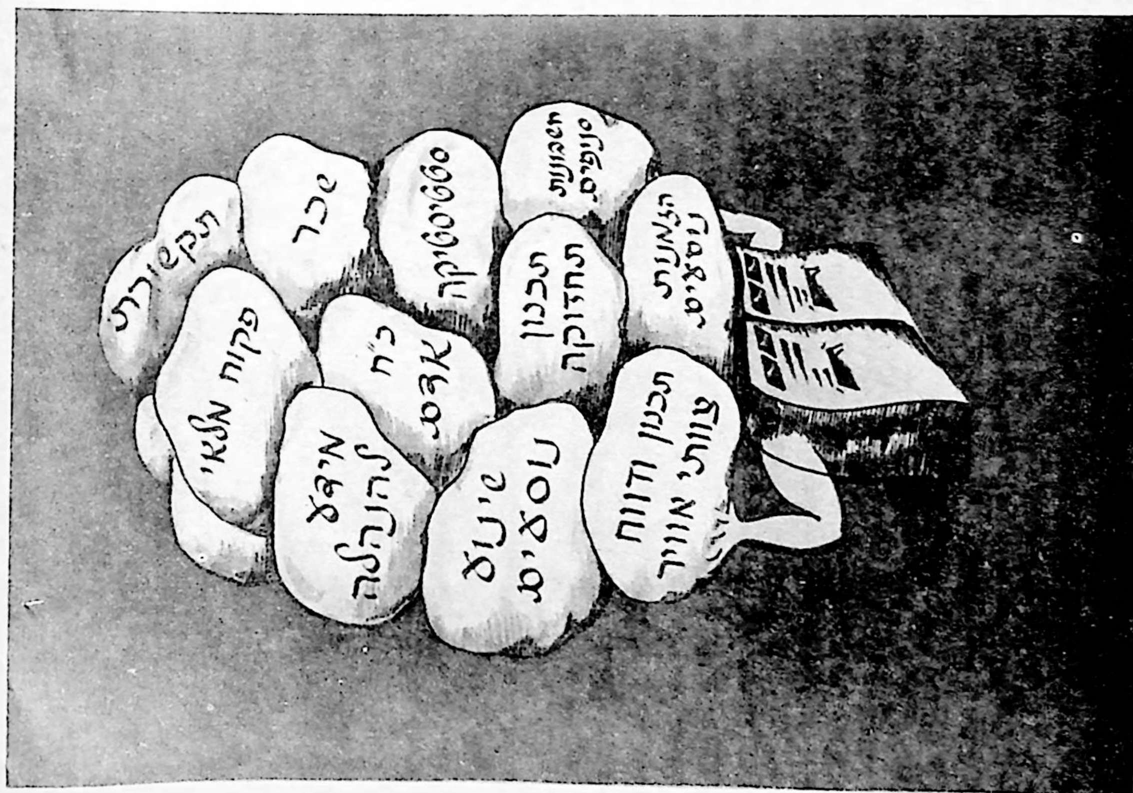
רשימת חברות התעופה המשתמשות באייפרס

- AER LINGUS
- ALITALIA
- AMERICAN AIRLINES
- ANSETT AIRLINES OF AUSTRALIA
- BRITISH AIRWAYS
- CANADIAN PACIFIC
- EL AL ISRAEL AIRLINES
- GARUDA INDONESIAN AIRWAYS
- JAPAN AIRLINES
- KLM
- KOREAN AIRLINES
- QANTAS
- PAN AMERICAN
- SINGAPORE AIRLINES
- SOUTH AFRICAN AIRWAYS
- SWISSAIR
- TAP
- THAI AIRWAYS

INTERNATIONAL
PROGRAMMED
AIRLINE
RESERVATIONS
SYSTEM

כרמל - ציוני דרך

* מינוי מינהלת הפרוייקט - ינואר 1974
* הזמנת ציוד - 1974/75
* תכנון והקמת הבנין - 1974/75
* לימוד אייפרס ע' עובדי מען והזמנות - 1975
* הצג הראשון פועל באולם המחשב - מאי 1975
* הדרכת עובדי ההזמנות מתחילה - ספטמבר 1975
* אולם המחשב החדש גמור - אוקטובר 1975
* הציוד מותקן - נובמבר 1975
* הרצה ונסיונות מתחילה - דצמבר 1975
* מערכת כרמל פועלת בתל-אביב - 18 בינואר 1976
* מערכת כרמל חוברה לירושלים - 27 בינואר 1976
* מערכת כרמל חוברה לחיפה - 29 בינואר 1976



כרמל
שינוע נוסעים

כרמל
הזמנות נוסעים

משקל ואיזון

מערכות מידע
וסטטיסטיקה

ניהול ופקוח
מלאי

תכנון ומעקב
תקציבי

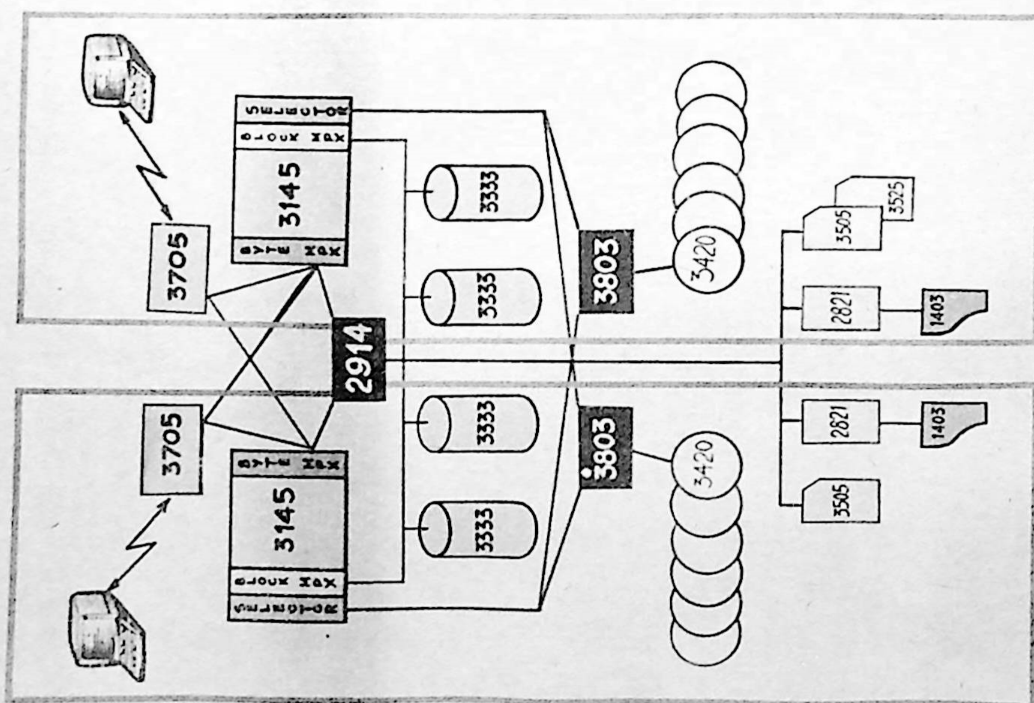
נתוח ביצועי
מטוס ומנוע

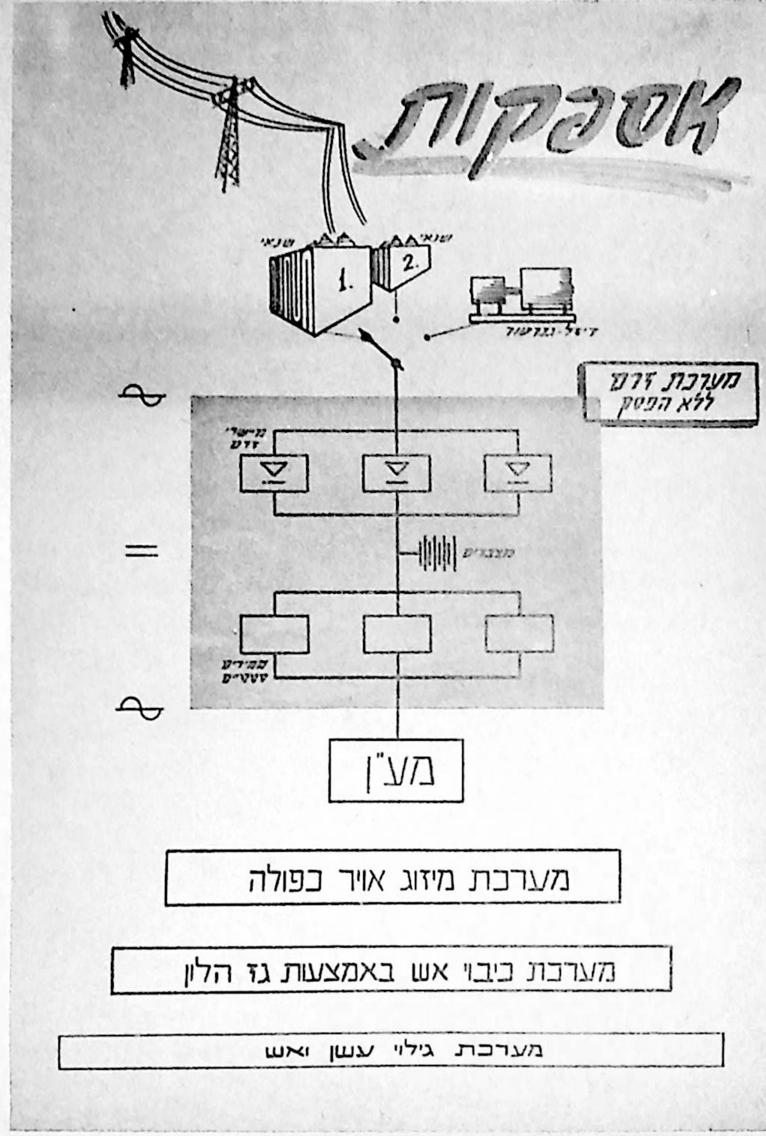
נהול השבונות

תכנון החזקת
מטוסים

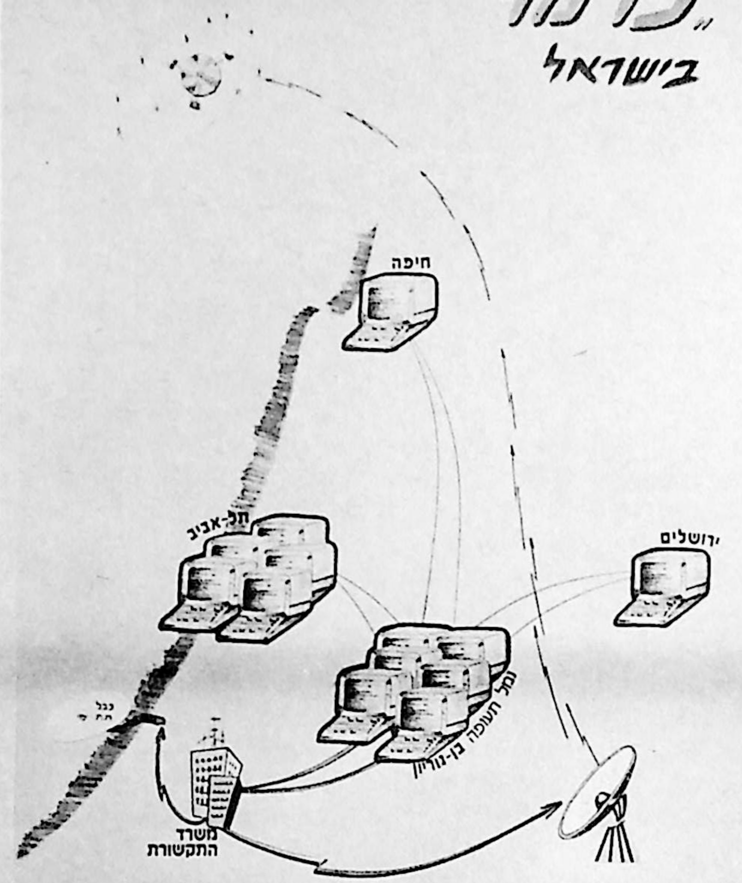
נהול כח אדם
ושכר

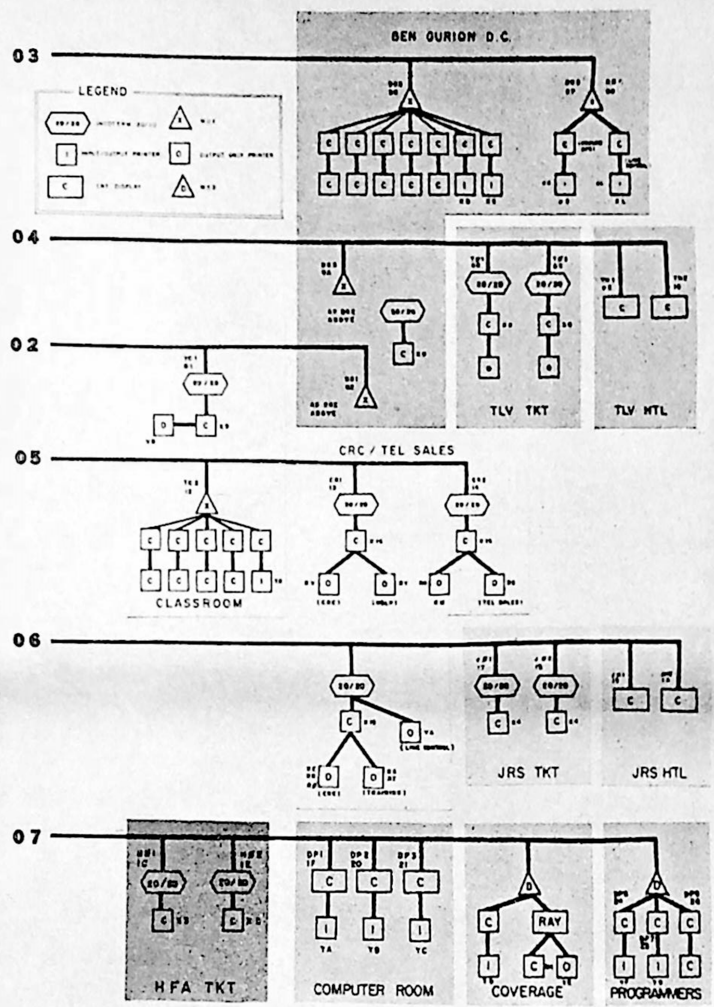
חקר ביצועים

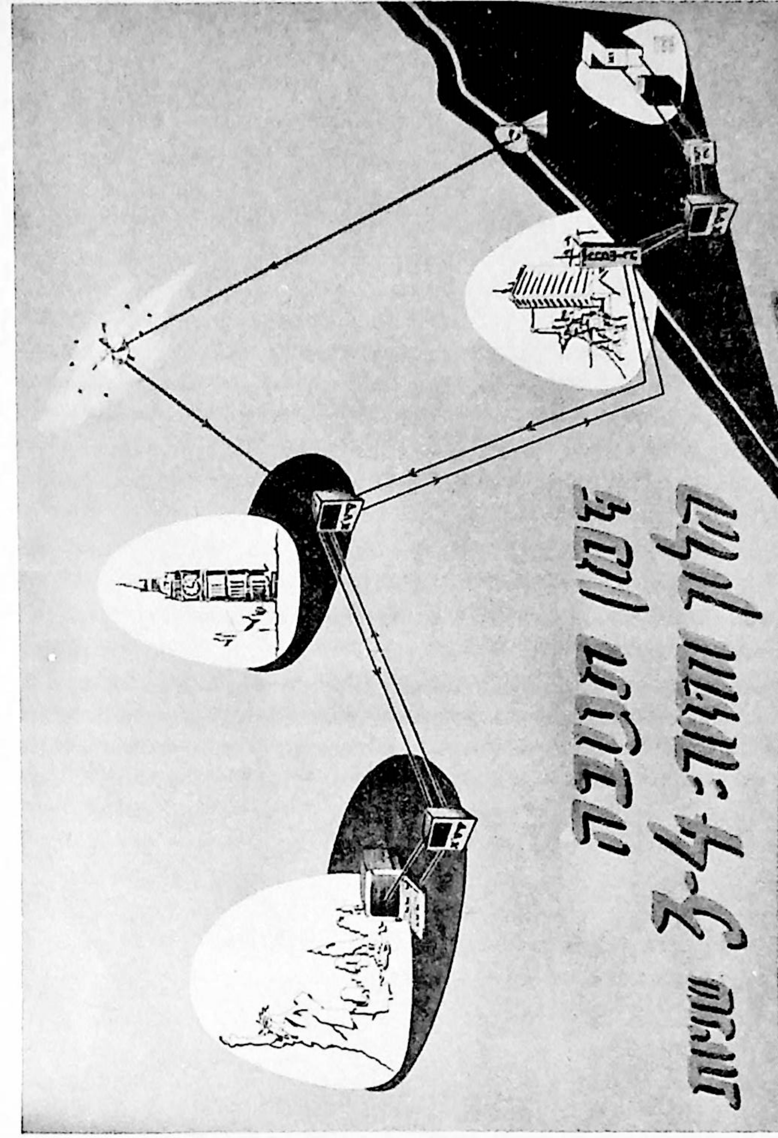




רשת התקשורת הבין-לאומית


מערכת
"כרמל"
בישראל
חיפה
תל-אביב
ירושלים
נמל תעופה בן-גוריון
משרד
התקשורת


EL AL
HIGH SPEED COMMUNICATIONS NETWORK
LEGEND
03
04
02
05
06
07
BEN GURION D.C.
TLV TKT
TLV HTL
CRC / TEL SALES
CLASSROOM
JRS TKT
JRS HTL
HFA TKT
COMPUTER ROOM
COVERAGE
PROGRAMMERS
RAY



מערכת כרמל

- משתמשת ב-800,000,000 בתי זכרון
- מטפלת ב-6 שאילתות לשניה
- מבצעת 98,000 פקודות לשניה
- משרתת כ3290 עם 36 מדפסות בארץ ובחו"ל
- מתפקדת 364 יום בשנה 24 שעות ביממה

רשת התקשורת -
אירופה, אפריקה ואוסטרליה
פירוט:
לונדון, פריס
מנצ'סטר, מרסיי
אמסטרדם, ניצה
וינה, ליאון
בריסל, סטרסבורג
ניקוסיה, אתונה
פרנקפורט, רומא
מינכן, מילנו
ברלין, בודפשט
קלן, קופנהגן
ציריך, שטוקהולם
ז'נבה, איסטנבול
טהרן, ניירובי
סידני, יוהנסבורג

רשת התקשורת - אמריקה
פירוט:
ניו-יורק, שיקגו
בלטימור, סנט-לואיס
בוסטון, אטלנטה
פילדלפיה, מיאמי
ושינגטון, יוסטון
פיטסבורג, לוס-אנג'לס
קליבלנד, סן-פרנציסקו
דיטרויט, מכסיקו-סיטי
מונטריאול, בואנוס-איירס
טורונטו, ריו-דה-ז'נירו

סימולצית מחשב של צומת ללא רמזור

יעקב ורול
אריה קויפמן
מיכאל חנני

אוניברסיטת בן-גוריון בנגב, באר-שבע

תקציר

בעבודה המוצגת כאן מובא מודל של תנועת רכב בצומת בלתי מרומזר ובסביבתו, ומתוארים שני יישומים של המודל. למשתמש במודל מאופשרת גמישות רבה בקביעת הנתונים הנוגעים למבנה הצומת, לטכניקת הבקרה של התנועה בו, לנפחי התנועה ועוד. ההדמיה של התנהגות הנהג במודל זה מבוססת על התייחסותו של הנהג לסביבה ועל הסקים פסיכו-פיסיולוגיים שהוא מסיק בהשפעת הסביבה. במודל זה לא נעשה שימוש בטכניקה המקובלת של התפלגות "המרווח המתקבל" (gap acceptance distribution), זאת כדי לאפשר שימוש בו למטרות של תכנון צמתים. עם זאת, בדיקות אימות הראו ש"מרווחים מתקבלים" המיוצרים על ידי מודל הסימולציה שקולים באופן סטטיסטי לנתוני שדה במידה ובתונים אלה נתנים למדידה. שני היישומים של המודל מבוססים על מודולריות מירבית, וכתובים בשפת התכנות פורטרן. יישום אחד מבוסס על הרצה במכלול (batch) בכל מחשב אשר יש בו קומפיילר פורטרן 4. היישום האחר הוא במיני-מחשב עם מערכת גראפיקה (graphics) ומאפשר עבודה אינטראקטיבית.

הקדמה

מהנדסי תנועה העוסקים באחזקה ובשיפור של צמתים קיימים, ואלה המתכננים צמתים חדשים, מתעניינים בתנועה בצמתים מבודדים. ענינם הוא לחזות מראש השפעה של שינוי בנפח התנועה, בטכניקות הבקרה בצומת ובמבנה הצומת על הביצועים של הצומת ושל הנהגים המשתמשים בו. יש מספר רב של עבודות בנושאי סימולציה של תנועה, ומאז שנת 1956 פותחו מודלים רבים המבוססים על שימוש במחשב לנושאים אלה (7). רב המודלים האלה עוסקים באופטימיזציה של הצבת רמזורים בצמתים של רשת דרכים, כגון (5), או שהם בנויים עבור צמתים מסויימים. מודלים בודדים מאפשרים ניסויים בגיאומטריות ובטכניקות בקרה שונות של צומת מסויים (1). במדינות מתפתחות, כישראל, מתכנני ערים נתקלים בבעיות של גידול בלתי חזוי של נפח תנועה בעקבות שינויים חריפים בגודל אוכלוסיה וברמת החיים. יתר על כן, מגבלות כספיות אינן מאפשרות תכנון לטווח ארוך, ויישומו. כתוצאה מכך יש צורך לשנות לעיתים מזומנות מבנה של צמתים קיימים ואת טכניקות הבקרה בהם, וגוברת החשיבות של זמינותה של מערכת סימולציה גמישה לעריכת ניסויים מוקדמים, בהקף מעבדתי, של שינויים כאלה.

המודל המובא כאן איננו מכוון לסוג מסויים של צומת. כמו-כן, למרות שהדגש במודל הוא על התנהגות הנהג (10), זהו מודל סימולציה של הצומת כלו, סביבתו וטכניקות הבקרה בו.

המוטיבציה להכנת המודל באה בחלקה ממהנדסי תנועה בישראל, הוא פותח ונכתב באוניברסיטת בן-גוריון בנגב ובמסר למשרד התחבורה (9) שמימן את הפרויקט. מפתחי המערכת בעזרו במדה רבה בדעות ובהצעות של מהנדסי תנועה ובעקר של אנשי מחלקת התנועה בעירית באר-שבע, בראשותו של מר יוסף שנהר.

הנחות כלליות

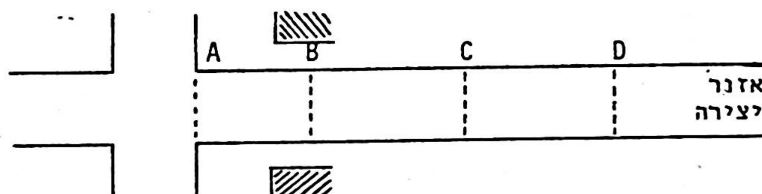
בהתבסס על נתונים אמפיריים (9) ובעקבות סקר של מחקרים בשטחים קרובים, התגבשו ההנחות הכלליות והפישוטים הבאים:

1. הצומת הוא הצטלבות של שני כבישים חד מסלוליים, דו כיווניים, עם ריבוי נתיבי תנועה רק בקרבה המידית של הצומת.
2. עקיפה אסורה בסביבת הצומת.

3. בקטעי הכבישים בסביבת הצומת אין כניסות של כבישים צדדיים.
4. קיימים במודל שני סוגים של מכוניות: גדולות וקטנות. נתונים שנאספו בשדה וראיו-נות עם נהגים מעידים על כך שלגדלים היחסיים של כלי הרכב יש השפעה רבה על החלטות הנהג.
5. קיימים במודל שני סוגי נהגים: נהגים "זהירים" ונהגים "פזיזים". הסיווג מבוסס על הערכותיהם של הנהגים את מצבי הסכנה בצומת ותגובותיהם להם. נהג זהיר פועל תמיד בהנחות קיצוניות להחמיר, ויהיה נוטה לתת זכות קדימה לנהגים אחרים, בעוד נהג פזיז מסתכן יותר ובוטה לקחת לעצמו את זכות הקדימה. סיווג זה מבוסס על ראיונות אישיים של נהגים ועל סיווגים שונים של נהגים כפי שהוצעו ב-(3).
6. בכבישי הגישה לצומת תנועתה של מכונית מאופיינת או כנסיעה חופשית (free motion) אם היא המכונית הראשונה, או כ"רכב עוקב" (car following) לשאר המכוניות. S. L. Cohen הראה ב-(2) שהשימוש ב"רכב העוקב" לתאור תנועה של מכונית, משפר את דיוק המודל בנפחי תנועה גדולים.
7. החלטה של נהג אם להכנס לצומת אם לאו מבוססת על התייחסותו לסביבה ועל הסקים פסיכו-פיסיולוגיים שהוא מסיק מכך. כדי להחליט, הנהג מעריך גדלים של פרמטרים דינמיים (כגון: מרחקים, מהירויות של מכוניות), בודק ומחשב מספר אפשרויות לפעולה תוך הנ-חות מסויימות באשר לתגובותיהם ולפעולותיהם של נהגים אחרים במערכת. ייצירת החלטה בדרך זו מאפשרת להמנע מהשימוש בהתפלגות של "מרווח מתקבל" כבסיס להחלטות של נהג בצומת. Wagner (11) וחוקרים אחרים ציינו מספר אי דיוקים הנובעים משימוש בהת-פלגות של "מרווח מתקבל". Wagner מדגיש את ההבדל בין תוצאות מחקריו ובין אלה של מחקרים אחרים בנושא, ומסיק שמקורם של הבדלים אלה הוא בכך שהתקבלו בצמתים שונים. לאור זאת, אם ברצוננו לערוך ניסוי בגיאומטריות, במבנים ובטכניקות בקרה שונות של צמתים, נושאים בהם בדרך כלל אין אפשרות לאסוף נתוני שדה, עלינו להמנע משימוש בפילוג של "מרווח מתקבל".

<u>תאור כללי של המודל</u>

המבנה הלוגי של המודל קשור בחלוקה, המלאכותית אך האפקטיבית, של קטע הכביש בקרבת הצומת, כמתואר בציור 1. מכונית נכנסת לצומת (יוצאת מהצומת) בחצותה את הנקודה A.



ציור 1: חלוקת כביש גישה.

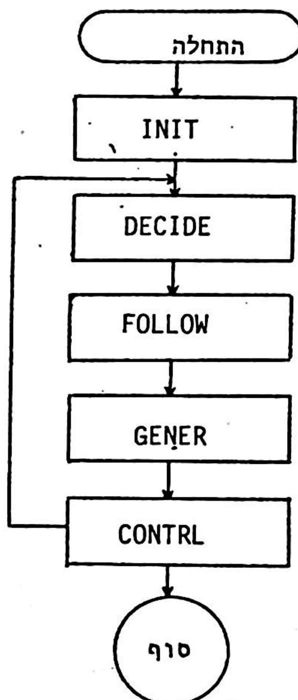
B היא הנקודה הרחוקה ביותר מהצומת בה יכול נהג לראות את כל כבישי הגישה האחרים למרחק BA לפחות. היינו, זו היא הנקודה אשר בה יכול נהג להחליט תוך בחינת אנלוגיה לעצמו, אם הוא נמצא בתנועה לקראת התנגשות עם מכונית אחרת. המיקום של הנקודה B הוא פונקציה של מספר גורמים, ביניהם: נתונים טופוגראפיים, כגון: עצים, בתים, עקומות בכביש, וכן תנאי מזג אויר וראות. הנקודה C היא הנקודה שבה נהג מתחיל להיות מושפע בשיקוליו מעצם קיומו של הצומת. זו היא הנקודה שבה נהג הנע במהירות ממוצעת מתחיל להאט בתאוטה קבועה נוחה (4) במטרה להעצר, אם יתעורר צורך בכך, לפני הצומת. הנקודה D יכולה להיות נקודה כל שהיא מעבר ל- C, בתנאי שאין כניסה של כביש צדדי בין D ובין הצומת. מכונית נעשית חלק אינטגרלי מ"המערכת המרכזית" כאשר היא חוצה את הנקודה D. לבסוף: מכוניות חדשות נוצרות ב"איזור יצירה" הנמצא מעבר ל- D, בהתבסס על יצירה אקראית של מרווח (אורך) בין מכוניות (headway).

תנועה מתבצעת במערכת בשיטת ה- Memorandum (או: שיטת הלגרנז'יאן) (8). הווה אומר: הזמן הוא בדיד, והמערכת כלה מתעדכנת בכל יחידת זמן (fixed time increment), כאשר הסיפול במהירויות ובמיקומים הוא רציף. אשר למבני אינפורמציה: לכל כביש גישה יש רשימה לינארית מעגלית (6) המכילה את המיקום, המהירות, סוג הרכב, סוג הנהג ותכונות נוספות של כל אחד מכלי הרכב הנמצאים בקטע AD באותו כביש גישה. סדר המכוניות בר-שימות האלה הוא הסדר שלהן למעשה בכביש. המכוניות הנמצאות ב"אזור הייצירה" מיוצגות בנפרד ברשימות לינאריות סדרתיות (6).

תכנית המחשב בנויה משישה מודולים נפרדים. המודול המרכזי אחראי על ההפעלה הסדירה של חמשת המודולים האחרים, והם:

התקנות ראשוניות	(initializations)	INIT
החלטת הנהג		DECIDE
תנועת הרכב		FOLLOW
יצירת מכוניות		GENER
בקרה		CONTRL

כמתואר בתרשים הזרימה הכללי בציור 2. בהפעלת התכנית, המודולים DECIDE, FOLLOW ו- GENER פועלים עבור כל אחד מארבעה כבישי הגישה לצומת. עקר פעולותיהם של המודולים מתוארים להלן:



ציור 2: תרשים זרימה כללי של המודל

INIT מודול זה מופעל פעם אחת בלבד בתחילת הסימולציה. ערכים לפרמטרים נקראים ומודפסים על ידו, נקבעים ערכים התחלתיים למשתנים ולמצבים, ונעשות הכנות להפעלת שגרות. כמו-כן, מאחר ובתחילת הריצה ריקים הכבישים ממכוניות, מודול זה מציב, באופן אקראי, מכונית אחת באזור היצירה של כל אחד מהכבישים.

DECIDE מודול זה עורך סימולציה של תהליך ההחלטה של נהג באשר לשינויים במהירותו (אם יש באלה) כפונקציה של מיקום הרכב בכביש, של התמרורים בצומת, של מגמת פניתו בצומת ושל נוכחותה של מכונית אחרת העלולה להיות בתנועה לקראת התנגשות איתו.

FOLLOW הוא המודול שבו מתבצע חישוב המהירות והמיקום של כל מכונית במערכת בכל צעד זמן, בהתבסס על שיטת "הרכב העוקב" ו/או בעקבות החלטות שהתקבלו במודול DECIDE.

GENER במודול זה נוצרת מכונית במערכת, בתחום שמעבר לנקודה D בכל אחד מהכבישים. מודול זה יוצר מרווח אורך בין מכוניות כדי לקבוע את מיקומה של המכונית הנוצרת, וכן הוא קובע את תכונות הנהג ורכבו. כמו כן מעדכן המודול הזה את כל כלי הרכב שנוצרו בעבר ואשר לא עברו עדיין את הנקודה D בכל כביש.

CONTRL במודול זה מעודכנים המיקומים והמהירויות של כל המכוניות במערכת בסופו של כל צעד זמן. מכיוון שההחלטות, הקידום של מכוניות והתגובות של נהגים במערכת תלויים אלה באלה, נעשים תחילה כל החישובים הרלונטים עבור כלי הרכב במודולים האחרים, ואילו עדכון מצב המערכת נעשה סימולטנית במודול זה. המודול שולח גם להדפסה תוצאות ביניים ותמונה סכימטית של הצומת. עם תם פעילויות אלה מקודם שעון הסימולציה ונבדק באם לסיים את הריצה או להמשיך בסיבוב נוסף. במדה ומחלט לסיים את הריצה, מודפסות תוצאות סופיות.

יצירת מכוניות

המודול GENER, שתואר בקצרה לעיל, נקרא לפעולה בכל צעד זמן, לאחר שהופעלו המודולים DECIDE ו- FOLLOW. עם הפעלתו של GENER, כמתואר בציור 3, הוא פועל בכל אחד מכבישי הגישה. כדי לאפשר ניסויים בצעדי זמן גדולים ובנפחי תנועה גדולים, המערכת יוצרת די מרווחי אורך על מנת לודא שהמכונית האחרונה בכביש לא תכנס למערכת המרכזית (הווה אומר: לא תעבור את הנקודה D) בצעד הזמן הבא. ייצירת מרווחי האורך נעשית תוך שימוש בפונקצית ההתפלגות:

$$P(h \leq x) = 1 - \exp(-\frac{x-L}{\bar{x}-L})$$

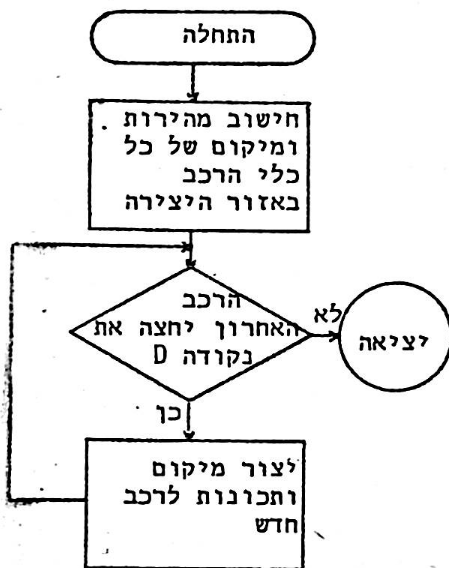
באשר h הוא מרווח האורך, L הוא אורך הרכב שלפני הרכב הנוצר, ו- $\bar{x}$ הוא מרווח האורך הממוצע. עקרונית, זהו פילוג אקספוננציאלי מוזז אשר מקובל להשתמש בו למטרות כגון זו, ולפיכך נוסה ואומת במספר רב של מחקרים (2). מהירויות התחלתיות של כלי רכב אשר נקבעות עם יצירת המכוניות, נשמרות כקבועות לפחות צעד זמן אחד, בערכים:

$$v = \bar{v}\frac{h}{H}$$

באשר: $H = 1+\frac{1}{\bar{x}+1}$ ($\bar{v}$ -המהירות הממוצעת, $\bar{x}$ -תוחלת מרווח האורך) וזאת במגבלות האילוץ שמהירות זו היא בין המהירות המינימלית לבין המהירות המכסימלית המותרות בכביש הגישה הנדון. סוגי מכוניות, סוגי נהגים ומגמות פניה בצומת נקבעים עבור המכונית הנוצרת בהתאם לאחוזיהם באוכלוסיה כפי שהם מוכתבים כפרמטרים של התכנית.

תנועת הרכב

בכל אחד מכבישי הגישה הטיפול בתנועת כלי הרכב נערך באופן אחד עבור הרכב המוביל (הרכב הקרוב ביותר לצומת) ובאופן אחר עבור כלי הרכב האחרים (הם כלי הרכב העוקבים). הסטטוס של רכב עוקב משתנה לזה של רכב מוביל כאשר הרכב שלפניו מתחיל להאץ במטרה להכנס לצומת. תנועת הרכב המוביל מבוססת על תנועה חפשית ועל אינטראקציה עם עצם קיומה של הצומת. תנועת כלי הרכב האחרים מבוססת על "תנועת הרכב העוקב" הידועה. עם זאת למעשה שני סוגי התנועה מבוצעים במערכת תוך שימוש באותן שגרות ובאותן פונקציות, המבוססות על "משואת הרכב העוקב". דבר זה נעשה על ידי דימוי הרכב המוביל לרכב עוקב הנע בעקבות מכונית דימיונית אשר מיקומה ומהירותה נקבעים באופן מלאכותי כדי לגרום לאפקט הרצוי על הרכב המוביל. מנגנון המכונית הדימיונית מאפשר לנו לאחד את החישובים הקשורים בתנועת כלי הרכב במודל. לדוגמא, אם ברצוננו להאט את מהירותו של הרכב המוביל במגמה של עצירה לפני הכניסה לצומת, אנו גורמים לרכב זה לנוע בעקבות מכונית דימיונית הנמצאת במצב עצירה בכניסה לצומת. כאשר על הרכב המוביל להאץ במטרה להכנס לצומת, הוא נע בעקבות מכונית דימיונית הנעה לפניו במהירות קבועה, גבוהה, הנקבעת בהתאם למגמת פניתו של הנהג בצומת ובתלות במהירות המכסימלית המותרת בכביש הגישה בו מדובר. תרשים זרימה כללי של תנועת הרכב במערכת נתן בציור 4.

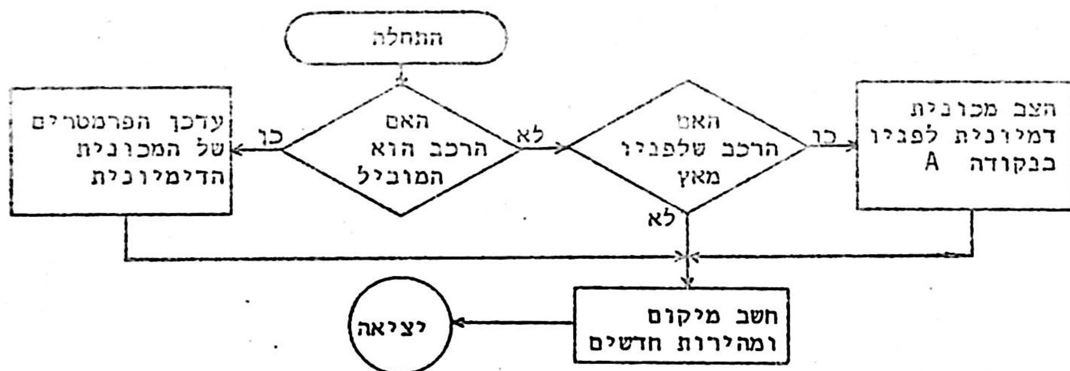


ציור 3: תרשים זרימה כללי של GENER

תנועת כלי הרכב מושתתת על המשואה הדיפרנציאלית הבאה, היא "משואת הרכב העוקב":

$$\ddot{x}_2(t) = (a+b\dot{x}_2(t))\frac{\dot{x}_1(t-T)-\dot{x}_2(t-T)}{x_1(t-T)-x_2(t-T)}$$

באשר $x_1(t)$ הוא המיקום של הרכב הראשון בזמן t, $x_2(t)$ הוא המיקום של הרכב העוקב לו בזמן t, ו- T הוא זמן התגובה הממוצע של נהג. הפרמטר b הוא קבוע מייצב, והפרמטר a תלוי ב- b וכן במהירות היחסית ובמרווח האורך בין המכוניות בזמן היצירה. המשואה נפתרת פתרון נומרי על ידי חלוקת צעד הזמן ל- 10 קטעים שוים, וחישוב המהירות והמיקום בכל אחד מ- 10 הקטעים (9). הערכים החדשים המתקבלים עבור המיקום והמהירות ב- 10 הקטעים של צעד הזמן הנוכחי נרשמים בתם החישוב באזור ארעי בזכרון, בעוד העדכון למעשה של תכונות כלי הרכב כתוצאה מחישובים אלה נעשה לאחר מכן בסופו של צעד הזמן, בבת אחת עבור כל המערכת. תוצאות החישובים האלה נשמרות באזור הארעי צעד זמן אחד בוסף, מאחר ונהג המכונית העוקבת של המכונית שעבורה נעשו החישובים יתייחס אליהם בצעד הזמן הבא - כתוצאה מהפיגור הנובע מזמן התגובה של הנהג.



ציור 4: תנועת הרכב

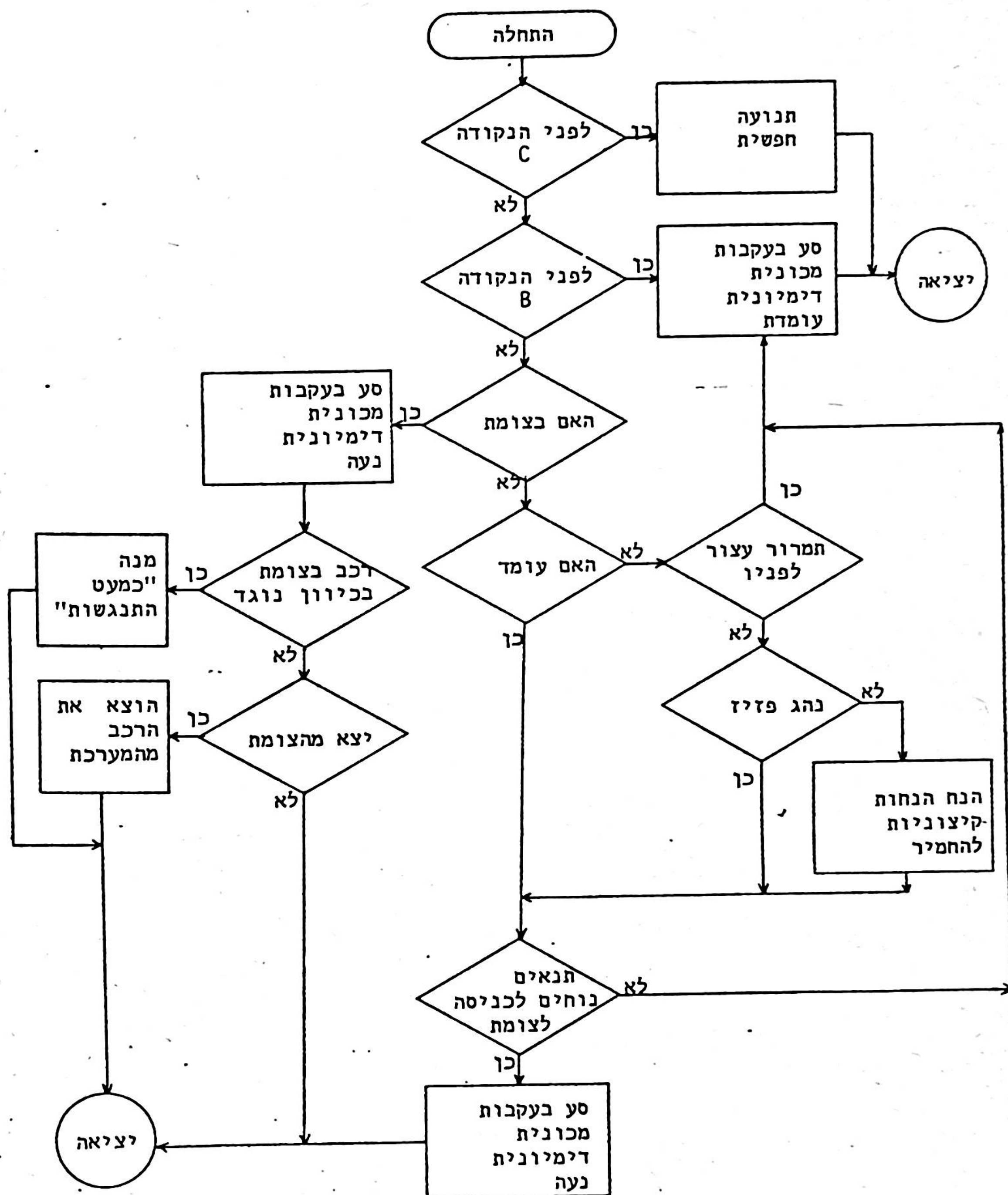
החלטות הנהג

המכונית המובילה (הראשונה) בכל כביש נעה בתנועה חפשית עד לנקודה C . בנקודה זו מתחיל הנהג להגיב לעצם קיומו של הצומת לפניו, כלומר: הוא מאט לקראת עצירה אפשרית לפני הכניסה לצומת. החלטה זו אינה משתנה עד לנקודה B , בה מתחיל הנהג המוביל להגיב לתמרורים בצומת ולכלי רכב אחרים הנמצאים בתחום של BA מהצומת בכל הכבישים. החלטות הנהג המוביל מתוארות בתרשים הכללי בציור 5.

אם נוכח המוביל יש תמרור "עצור", אין הוא משנה את החלטתו האחרונה וממשיך בתאוטה לקראת עצירה. אחרת - הנהג מתחשב בכלי רכב אחרים הנמצאים בקרבת הצומת, ובהתבסס על תכונותיהם ועל תכונותיו הוא מחליט באם להמשיך בתאוטה או להאץ במטרה להכנס לצומת. המודל מבחין בין נהג פזיז ובין נהג זהיר באשר לקביעת הרגע בו תפול ההחלטה להאץ. נהג פזיז עשוי להחליט בנקודה B להאץ במטרה להכנס לצומת, בהתבסס על הערכתו במדוייקת, ולפעמים המוגזמת באשר ליכולתו, את מצב התנועה. לעומת זאת, נהג זהיר ידחה את החלטתו להאץ וימשיך להאט גם אחרי עברו את הנקודה B . נהג כזה מניח שיש רכב, בכביש אשר לתנועה בו עליו לתת זכות קדימה, הנמצא מיד מעבר לטווח ראייתו והנע במהירות מכסימלית.

הגורם המכריע המשפיע על החלטת הנהג המוביל באשר לכניסה לצומת הוא המטרה להמנע מהתנגשות עם כלי רכב אחרים בין בתוך הצומת ובין, התנגשות מאחור, בנתיב היציאה מהצומת. בתחילה המודל מחשב אפשרות של התנגשות כתלות במגמות הפניה בצומת ובתמרורים שבצומת בלבד. המודל מניח שמגמות הפניה בצומת של כל כלי הרכב וכן כל התמרורים המוצבים בצומת ידועים לנהג המוביל המחליט בעת הגיעו לנקודה B . ברור כי לא כל כלי הרכב הנמצאים בסביבת הצומת מהוים סכנה של התנגשות, אולם אם מגמות הפניה של כלי הרכב ומערך התמרורים גורמים להווצרות מצבים העשויים להוביל להתנגשות, הנהג המחליט מעריך את זמני ההגעה של כלי הרכב למספר נקודות מפתח, ומבסס את החלטתו על תוצאות הערכות אלה. בחישוב הערכות אלה אנו מניחים שהנהג המחליט יודע מהי מהירות רכבו, ומכיר בדיוק את תכונותיו הדינמיות. לעומת זאת הוא מעריך הערכה שאינה מדוייקת אורכים של קטעי דרך ומהירויות של כלי רכב בכבישי הגישה האחרים אל הצומת. אחד הכלים החשובים בהם משתמש הנהג בהערכותיו הפסיכו-פיסיקליות באשר לכלי רכב אחרים הוא: קצב השינוי של זוית הראיה שלו. מהירויות ומרחקים מוערכים על ידי שיגרה המשתמשת בפילוג נורמלי קטוע, ואשר משתמשת בתוצאות המובאות ב- (3), וכן מתחשבת בסוג הנהג, בתנאי הראות ובפרמטר המציין מצב של יום או לילה.

גם כאשר מחלט להאץ, הנהג המוביל ממשיך לשים לב לתנועה בכבישי הגישה האחרים כשהוא מוכן לשנות את ההחלטה באם תתעורר אפשרות של התנגשות פוטנציאלית אשר לא הורגש בה קודם לכן. במקרה שאפשרות כזו מתעוררת, הנהג מאט מיד בתאוטה גבוהה כשמטרתו: מניעת התנגשות. אם המכונית בכל זאת נכנסה לצומת - התכנית רואה במצב זה מצב של "כמעט התנגשות" ועורכת רישום מתאים לצורך הדיווח הסטטיסטי. כאשר נוצר תור של מכוניות עומדות לפני הצומת, המכונית המובילה נתקת מהתור ונכנסת לצומת בעקבות אותם מהלכי החלטה שתוארו לעיל עבור מוביל בתנועה. כניסת המוביל לצומת גורמת לכך ששאר המכוניות הממתינות בתור יאצו במטרה להתקדם בתור בכיוון הצומת



ציור 5: החלטות נהג

יישום המודל במחשב

המודל הוכן להרצה במחשב כתכנית סימולציה בשפה פורטרן, תוך שימוש בשיטת צעד זמן קבוע. הסיבות להעדפה של שפת התכנות הכללית פורטרן על פני שפות תכנות המיועדות לסימולציה, כגון SIMULA או SIMSCRIPT היו:
א. מכיון שהמודל מבוסס על שיטת צעד הזמן הקבוע, היתרונות של שפות הסימולציה לא בלטו.
ב. תכנית סימולציה של תנועה בשפת פורטרן חסכונית בזמן ובמקום בזכרון (12).
ג. פורטרן מקובלת יותר, וזמינה ברובם הגדול של המחשבים, בעקר אצל מהנדסים ומוסדות העוסקים בבעיות תעבורה.
התכנית נכתבה באופן מודולרי על מנת להקל על הרחבתה בעתיד ולאפשר הכנסת שינויים כפי שעשויים להדרש על ידי מהנדסי התעבורה.

התכנית קיימת בשתי מהדורות. האחת היא תכנית ייצור, הפועלת במחשבים בעלי 32K מילות זכרון לפחות ולהם קומפיילר של פורטרן 4. (הדגש לא הושם על נצול יעיל של זכרון כדי להמנע מתלות במחשב מסוים). הפרמטרים הנמסרים לתכנית בקלט בתחילת ההרצה הם קבועים ואינם נתנים לשינוי בזמן הריצה. פלט, בין פלט סטטיסטי ובין שרטוטים של הצומת, מתקבל במדפסת מהירה. הרצות הניסוי נעשו במחשב CDC CYBER 73 תוך שימוש בקומפיילר FTN (ללא אופטימיזציה). זמן הקומפילציה: 10 שניות (כ- 1200 משפטי פורטרן). יחס ממוצע של 10:1 התקבל בין זמן דימוי (זמן סימולציה) לבין זמן אמיתי עבור צומת המשרת כ- 1000 כלי רכב לשעה. דוגמא לפלט חלקי מובאת בציור 6. המהדורה השניה היא תכנית המבוססת על חמישה רבדים (overlays) והמותאמת להפעלה אינטראקטיבית במחשב קטן בעל זכרון של 24K מילים לפחות, של 16 ביט. שימוש במצג גראפי, בעט-אור (light-pen) ובלוח מקשים מאפשר אינטראקציה עם התכנית בעת הריצה. קטע התכנית השוכן דרך קבע בזכרון הוא המודול המרכזי, וחמשת הרבדים תואמים את חמשת המודולים של המערכת כמתואר בציור 2. המהדורה הזו מותקנת במחשב DEC PDP 11/40 , ויחידת הגראפיקה היא מסוג GT-44 . קומפילציה וקישור (linking) אורכות כ- 10 דקות במערכת ההפעלה DOS/BATCH . היחס: זמן סימולציה כנגד זמן אמיתי הוא בממוצע 1:9. "תנועה איטית" זו מקורה במבנה הרבדים של התכנית, והיא מאפשרת אינטראקציה נוחה של המשתמש עם התכנית בפעולתה. תמונת הצומת המופיעה על המסך הגראפי דומה לזו שבציור 6.

אימות המודל

במהלך פיתוח ויישום המודל, המודולים הבלתי תלויים וההנחות העקריות עברו תהליך של בחינות אימות. נושאי ההתפלגות למרווח בין מכוניות והשימוש ב"רכב העוקב" נבדקו ותוצאות המבחנים אישרו את הסיכומים המופיעים ב- (2). אימות כללי של המודל בוסס על ניתוח "המרווח המתקבל" ומספר ממוצעים כלליים, כגון: זמן שהיה של רכב במערכת או מספר כלי הרכב שטופל בהם ביחידת זמן. התוצאות אישרו שמודל החלטות הנהג לא סותר את התפלגות "המרווח המתקבל" כאשר היו נתוני שדה ליצור התפלגות כזו. הנתונים נאספו הן מצומת ארבע-כיווני והן מצומת תלת-כיווני. ערכים התחלתיים ונתוני אימות נאספו בו-זמנית בשיטה הדומה לזו של Wagner (11). תהליכים לא-פרמטריים כגון Wilcoxon Test וכן Kolmogorov-Smirnov Test יושמו לבדיקת ההנחות. אחד הצמתים הקטלניים בבאר-שבע תוכן מחדש על ידי מחלקת התחבורה בעיריה, והתוצאות יושמו בשטח. תכנית הסימולציה נוסתה גם היא בעזרת הנתונים שנאספו במסגרת התכנון הזה, והתוצאות שהתקבלו תאמו את העובדות בשטח, דבר המבסס ביסוס נוסף את מהימנות המודל.

מגמות לפיתוח נוסף

פיתוח נוסף של המערכת כולל הרחבת האינטראקציה במהדורה השניה של התכנית במטרה לאפשר עצירת הריצה, שינוי פרמטרי קלט, שינוי תכונות של כלי-רכב ואף הכנסת כלי -רכב חדש למערכת. בנוסף לשימוש בתכנית למטרות תכנון, נראה שנתן להשתמש במודל להשתלמויות נהגים. המהדורה האינטראקטיבית מתאימה במיוחד למטרה זו, כאשר שליטה מלאה על כלי-רכב מסויים נתונה בידי הנהג המשתלם באמצעות עט-אור, וכאשר שאר כלי הרכב במערכת נשלטים על ידי המודל. מספר נסיונות נדרשים כדי לכוון את הנהג לפענוח נכון של המידע המופיע על המסך, אולם כאשר מתגברים על קשי ראשוני זה, מוצגים לפני הנהג מצבי התנגשות שונים שבהם הוא צריך ליישם את כישוריו.

CLOCK = 33

(Y=YIELD SIGN ...)

2.NORTH: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
.22	-1	1	0
1.67	1	2	0
2.90	0	2	1

1.EAST: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
3.40	0	1	0

3.WEST: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
2.10	-1	1	1
3.27	0	1	0

4.SOUTH: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
3.73	0	1	0
1.79	0	2	1
2.75	0	2	0

ציור 6: תרשים הצומת המהווה פלט חלקי
של התכנית

ספרות

(1) Ashworth, R., Goodwin, R.F., and Cheung, N.C. Simulation of traffic relays at a T-junction with various alternative forms of layout. SIMULATION, Sept. 1975, 145-148.

(2) Cohen, S.L. Application of simulation network models to the analysis of urban intersection performance. Final Report, Traffic Systems Division, Office of Research, Federal Highway Administration, U.S. Department of Transportation, 1972.

(3) Forbes, T.W. (ed.) Human Factors in Highway Traffic Safety Research. John Wiley and Sons Inc. 1972.

(4) Highway Capacity Manual. Highway Research Board Special Report 87, 1965, Washington D.C.

(5) Kaufman, A.A. Computer simulation and synchronization of traffic signals. Proc. 2-nd Jerusalem Conference on Information Technology 1974, 129-135.

(6) Knuth, D.E. The Art of Computer Programming, vol. 1. Addison-Wesley 1969.

(7) Report of Road Research Group on "Area Traffic Control Systems" 1972. Organization for Economic Cooperation and Development, Paris.

(8) Solomon, A., Hanani, M.Z., Varol, Y.L., and Shapiro, J. A simulation program for a signal free intersection of two roads. Technical Report Math-10, Ben-Gurion University of the Negev, Beer-Sheva, Israel 1972.

(9) Solomon, A., Harband, J., Varol, Y.L., Hanani, M.Z., Kaufman, A.A., and Shapiro, J. Simulation of a signal free intersection. Final Report, submitted to the Ministry of Transport, Israel 1975 (in Hebrew).

(10) Varol, Y.L., Kaufman, A.A., and Hanani, M.Z. Driver-vehicle behaviour at an intersection: a model for intersection design. SIMULATION (to appear).

(11) Wagner, F.A. Jr. An evaluation of fundamental driver decisions and reactions at an intersection. Highway Research Record 118 1966, 68-84.

(12) Weinert, A.E. A Simscript-Fortran case study. Comm. of the ACM vol. 10 (1967) 784-792.

CLOCK = 33

2.NORTH: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
.22	-1		2
1.67	3		2
2.96	3		1

(Y=YIELD SIGN , S=STOP SIGN)

1.EAST: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
3.40	0	1	0

3.WEST: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
3.15	-1	1	1
3.27	0	1	1

4.SOUTH: VELOCITY TURN TYPE-VEHICLE TYPE-DRIVER

VELOCITY	TURN	TYPE-VEHICLE	TYPE-DRIVER
3.71	2	1	2
1.79	3	1	1
2.35	6	2	1

ציור 16 תרשים הצומת, המהווה פלט חלקי של התכנית

232

שמושי מחשב בתחבורה ימית־נסיון "צים"

פרופיל חברת צים

דוד רוזן

1. אניות בבעלות - 65

2. אניות בהפעלה - 120–150 (בעלות + הבורות)

3. נפח הצי - 2.5 מיליון טון

4. עובדי ים וחוף - 4000

5. סניפים - 38

6. סוכנים - 250

7. נמלים בפקידה - 200

8. מחזור כספי - כ-400 מיליון דולר

9. קווי ספנות עקריים:- - א. קו שלוש יבשות;
   ב. קווי אילת;
   ג. קוו ים תיכון אירופה;
   ד. קווי כוכב הזהב.

מאפייני ענ"א בחברת "צים"

1. פיזור גאוגרפי -

פעילות בכל העולם. יחידות וציוד במרכזי עבוד רחוקים הפועלים בצורה עצמאית במסגרת מתואמת.

2. אילוץ זמן -

הולך וגובר עם הגברת מהירות האניות וקצור שהיות.

3. תהכום המידע -

פיזי - כתוצאה מהמכלה היוצרת מספר רב של משתתפים הקשורים וחלויים זה בזה.

כספי - כתוצאה מהנ"ל וכמויות הולכות וגדלות.

4. הקשורת -

העברת הנתונים ממרכז למרכז ולפריפריות בצורות שונות:-

- בקווים חכורים לנתונים;
- בקווי חיוג ;
- בקווי טלקס;
- באפיקים חכורים (טלפון, נתונים טלקס משולבים)

מ ר כ ז י   ה ע ב ו ד

מספר הרשת	המרכז	לאזור	ציוד
1.	חיפה	ישראל מזרח קרוב משרד ראשי	IBM 370/135 2 x PDP - 15
2.	גנואה	איטליה צרפת ספרד צפון אירופה	PDP - 15
3.	ניו-יורק	מזרח ארצות הברית מערב " " קנדה	2 x PDP - 15
4.	טוקיו	יפן קוהיאה טיוון הונג-קונג סינגפור פיליפינים	1 x PDP - 15

תיאור סכימטי של מחזורי נתוני הפלגה

1. מחזור היצוא

1.1 מסד הנתונים

שיוכיים
- שולח
- מקבל
- בנק
- כתובות

כספיים
- סוג
- תעריף
- חישובים
- סוגי תשלום

פיסיים
- כמות
- אריזה
- משקל
- נפח
- מכולה
- סימנים

שטר מטען

יחידת מידע בסיסית לכל התהליכים.

1.2 עבוד נתונים

עבוד נתונים

גופים פנימיים

תפעול-שיווק-הנה"ח
סוכנות

גופים חיצוניים

מכס - נמל- לקוחות
מובילים

העברת נתונים

2. מחזור היבוא

עבוד נתונים

גופים פנימיים

הנה"ח - שיווק - תפעול
סוכנות

גופים חיצוניים

מכס - נמל - לקוחות
מובילים

מאפיינים

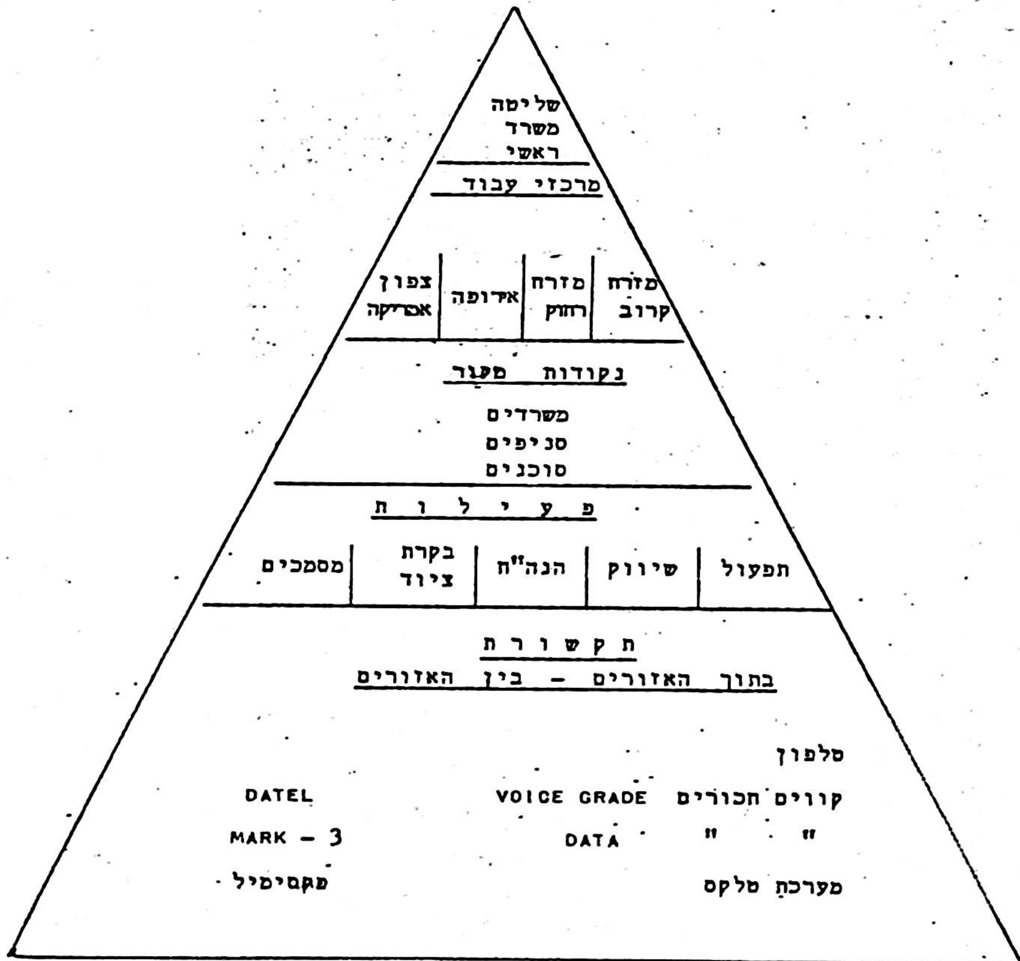
איסוף - עבוד - העברה - עבוד - פיזור

אילוצים

זמן - מרחק - כמות - מספר רב של גורמים בתהליך.

- טפנס

פירמידת הפעילות



מערכת הפעלה ON – LINE

M	U	M	P	S
MASSACHUSSETTS' GENERAL HOSPITAL	UTILITY	MULTI	PROGRAMMING	SYSTEM

תיאור טכני

1. מערכת הפעלה ושפה אינטגרלית.

2. השפה – INTERPRETER נמצא בזכרון.

3. הקצאת מחיצה לכל משתמש בגודל קבוע. התכנית והמשתנים נשארים במחיצה. ניתן לשנות גודל מחיצות בתכנית.

4. שתוף זמן ע"י TIME SLICE קבוע.

5. PRIORITY לגמר פעולת I/O.

6. MEMORY DISK BUFFER POOL לצמצום הגישות לדיסק.

7. מבנה קובץ במתכונת עץ בינרי עם רשומות באורך משתנה.

8. גישה מבוקרת לכל המשתמשים לקבצים בו-זמנית.

9. מערכת שגרות שרות לאחזקה שוטפת.

MUMPS

מ א פ י י נ י ם

1. קל ללמידה - קורס בן שבוע.

2. אין כלל צורך לידע מפורט ברמת מערכת.

3. <u>כתיבה מהירה</u> - כתיבה ON-LINE
   - ניפוי ON-LINE
   - ניסוי ON-LINE
   - שינוי בעת עבודה
   - העתקת קטעי תכנית.

   ON-LINE

4. <u>הפעלה נוחה</u> - על-ידי מסוף מדפיס.
   - על-ידי מסוף מסך.
   - כתיבה על סרט.

   ON-LINE

5. כתיבה
ניסוי
הפעלה
הזנה



הקמה מהירה
שינוי קל
הפעלה ע"י משתמשים
מילוי דרישות מהיר
עלות נמוכה.

שביעות רצון
- צרכנים
- עובדי ענ"א

מ ד י נ י ו ת   פ ת ו ח   כ ל ל י

1. ביזור מרכזי עבוד.

2. שמירה על שפה אחידה במידת האפשר.

3. שילוב מערכות: טלקס ;
נתונים ;
טלפון דבור ;
מערכות עבוד.

4. העברת נתונים סלקטיבית.

5. פתוח - הדרגתי;
- מתואם;
- מקביל.

ש א ל ו ת   -   ת ש ו ב ו ת

28.7.1976

מודל לתכנון תובלה אווירית, ימית ויבשתית

של "אגרקסקו"

א. בן-כוזרי

מבוא.

"אגרקסקו" היא חברה ליצוא חקלאי, המיצאת תוצרת חקלאית טריה לארצות מערב אירופה.

החברה משתמשת באמצעי תובלה מגוונים, של מטוסים חכורים, טיסות קו, אניות חכורות ואניות טרמפ, על מנת להעביר את התוצרת מן הארץ למרכזי השטעון באירופה, ומשם הפזור נעשה באויר או ביבשה עד ליעדים הסופיים.

בגלל נפחי התובלה הגדולים, על החברה להכין מבעוד מועד את אמצעי התובלה שיעמדו לרשותה בעונת היצוא. כיוון שרשת השווק של החברה רחבה ומסועפת ואמצעי התובלה רבים ומגוונים, הוחלט לבנות מודל לתכנון התובלה, שהוא מודל בתכנון ליניארי, אשר מסוגל להעריך ולהציע את תכנון אמצעי התובלה לעונה, שבוע אחר שבוע. המודל כולל במערכת אחת את התובלה האוירית, הימית והיבשתית מישראל לאירופה ובתוך אירופה סימולטאנית.

המודל היה דרוש על מנת לעזור לצוות התכנון באגף התובלה של החברה, לנתח את כל האינפורמציה כדי להגיע להחלטות נכונות.

התוצאות המתקבלות מן המודל ממליצות על:

1. סוגי המטוסים והאניות שיש להשתמש בהן.
2. מס' הטיסות וההפלגות שיש לבצע מידי שבוע.
3. שדות התעופה או הנמלים אליהם כדאי להגיע.
4. אמצעי התובלה שיש להשתמש בהם באירופה לפיזור התוצרת (אם אלו משאיות או רכבות).
5. עלות של מערכת זו.

תאור המודל

כללי:

מערך השווק של "אגרקסקו" באירופה בנוי כרשת אשר מקורה בארץ בתחנות היצוא נתב"ג - בדרך האוויר, ונמלי אשדוד וחיפה - בדרך הים.

הצמתים מפוזרים באירופה ונקראים מרכזי שטעון בנמלי ים ואויר גדולים כמו ציריך, פרנקפורט, פריס ולונדון באוויר, ונמלי מרסיי ורוטרדם בים.

היעדים הסופיים (Sink) הם סניפי החברה בארצות השונות, סניף בכל מדינה. (מבנה הרשת נספח 1).

ההצגה המתימטית של המודל .

1) $\sum_i \sum_l \sum_k X_{mjlki} \geq D_{mj}$

מגבלות הביקוש למוצר מסויים בצומת מסויימת .

2) $\sum_i \sum_m X_{mlki} \leq W_{lk} * n_{lk}$

מגבלות משקל לקטע מסויים בכלי הובלה מסויים .

3) $\sum_i \sum_m X_{mlki} * \rho_m \leq V_{lk} * n_{lk}$

מגבלות נפח לקטע מסויים בכלי הובלה מסויים .

4) $\sum_i \sum_k \sum_l X_{mjlki} - \sum_{i \neq j} \sum_k \sum_l X_{mjlki} = \sum_l \sum_k X_{mjlki}$

מאזן צומת מטעון למוצר מסויים.

5)

$$n_1 - I_0 \geq 0$$

$$n_r - I_r U_r \geq 0 \quad r \neq 1$$

$$n_r \leq I_{r-1} U_r \quad \forall r$$

$$\eta_r - I_{r-1} = 0 \quad \forall$$

$$n_r \geq U_r$$

$$I_r = 0 \text{ or } 1$$

$$\sum_{r=1}^{l} \eta_r = M_{k'}$$

n_r - מס' טיסות במטוס r ממטוסים מסוג k'

U_r - חסם עליון למס' טיסות במטוס r

η_r - מס' המטוסים שנחכור , η_1 - המטוס הראשון, η_2 - המטוס השני וכך הלאה.

I_r - משתנה החלטה האם נשתמש במטוס אם לאו .

המשוואות הנ"ל מגדירות יחס של עדיפות בין המשתנים , לא נשתמש במטוס ה- η_r כל עוד המטוס η_{r-1} לא הגיע לחסם העליון שלו דהיינו U_{r-1}.
זה מבטיח לנו שכל מטוס שנחכור יבצע תחילה את כל הטיסות המורשות לו ורק אח"כ נחכור מטוס שני .

6) $J_{i,s} Q_{i,s} \geq P_{i,s} \quad s \neq i$

$$J_s Q_{i+1,s} \geq P_{i+1,s} \quad s \neq i$$

$$\sum_{i>2} P_{i,s} \leq P_s \quad s \neq i$$

$$P_{i,s} \leq Q_{i,s}$$

P_1 - מס' הכניסות ליעד מסויים חסום מלעיל ע"י

$P_i, i \geq 2$ - מס' היציאות מאותו יעד ל i יעדים שונים.

J_1 - משתנה החלטה האם לנחות ביעד הראשון .

$J_i, i \geq 2$ - משתנה החלטה האם להמשיך לנחיתה כפולה .

Q_i - חסם מלעיל למספר הכניסות והיציאות .

המשוואות הנ"ל מבטאות את התנאים הבאים:

א. אם $J_1 > 0$ אז $J_i \geq 0, \forall i \geq 2$

ב. אם $J_1 = 0$ אז $J_i = 0 \quad \forall i$

פונקציית המטרה

$$\text{Min}\left\{\sum_{l'}\sum_{k'}\left(C_{k'}I_{k'}M_{k'}+b_{l'}J_s\rho_s N_{k'}\right)+\sum_{\bar{k}}\sum_{\bar{l}}X_{m\bar{l}\bar{k}}*\Psi_{m\bar{l}\bar{k}}\right\}$$

m = מוצרים .

j = צומת .

l = קטע .

k = סוג תובלה.

i = יעד סופי.

W = מגבלת משקל של סוג תובלה k בקטע l .

N = מס' נסיעות/טיסות/הפלגות של סוג תובלה k בקטע l.

ρ_m = יחס נפח/משקל התלוי במוצר m .

V = מגבלת נפח של סוג תובלה k בקטע l.

ψ = מחיר טון הדבלה שהוא פונקציה של סוג התוצרת, סוג ההובלה והקטע .

M = מס' כלי תובלה חכורים .

$\bar{I}$ = {קטעים בהם משתמשים בשרותי קר} /l

l' = {קטעים בהם מופעלים כלים חכורים} /l

$\bar{k}$ = {כלי שרות קר} /k

k' = {כלים חכורים} /k

מן ההצגה המתימטית של המודל אנו למדים , מדוע שיטת הפתרון למודל הנ"ל צריכה להיות תכנון מעורב של תכנות ליניארי ותכנון בשלמים –
"mixed integer programming" .

א. חלק מהמשתנים במודל מייצגים גדלים שאינם ניתנים לחלוקה, לדוגמא: מטוסים ואוניות חכורות וכאן יש משמעות לגודל הבדיד השלם של המשתנה.

ב. הצורך בשימוש במשתני החלטה לדוגמא:
טווח העבודה של המודל הוא שבוע , אחת ההחלטות שהמודל עוזר לנו לקבל היא , כמה מטוסים או אוניות על החברה לחכור על מנת לעמוד באילוץ כמויות התוצרת , שעומדות להעברה לאירופה . כאשר לכל מטוס יש מגבלה של 8 טיסות שבועיות, כלומר עלי להשתמש במשתנה החלטה , שיאלץ לנו מטוסים נוספים ברגע שהמטוס הקודם היגיע למגבלת הטיסות שהוא מסוגל לבצע, ולחברה עדיין לא העבירה את כל הכמויות לאירופה בשבוע הנדון .

ג. מבנה פונקציית המטרה שמרכיביו הם : הוצאות קבועות והוצאות משתנות התלויות בפעילות .

כל הגורמים הנ"ל הצביעו על פתרון mixed integer programming .
כאשר הכלי הוא M.P.S.X + M.I.P של I. B. M. .

כעת נפנה לתאור הבעיה הספציפית והיא בעיית תכנון התובלה של "אגרקסקו" שבאה על פתרונה בשימוש במודל שתואר .

3. יישום המודל.

למעשה עומדות בפני "אגרקסקו" מידי שנה בשנה שתי בעיות עקריות:

א. כיצד לבנות תחזית על בסיס שבועי של הקצאת תוצרת לשווקים השונים באירופה על מנת לקבל בהם פדיון מכסימאלי.

ב. כיצד להעביר את התוצרת החזויה על בסיס שבועי ליעדים השונים באירופה כך שהוצאות ההובלה תהיינה מינימאליות.

ברור כי שתי הבעיות חד הן ותלויות זו בזו.
אפשר למצוא פתרון אופטימאלי עבור בעיה א' בהינתן פתרון לבעיה ב' ולהיפך, ניתן לפתור אופטימאלית את בעיה ב' בהינתן פתרון לא'.
במקרה שלפנינו אנו עוסקים בפתרון אופטימאלי של בעיה ב' בהינתן הפתרון ל א', ובהזדמנות אחרת ניתן להציג כיצד אנו פותרים את בעיה א' בהינתן הפתרון ל ב'.

3.1 ובכן כפי שציינתי אנו יוצאים מתוך הנחה כי עומדת לפנינו התפלגות כמויות התוצרת לפי שבועות על פני העונה כולה, כאשר בכל שבוע נתונה לנו התפלגות הכמות השבועית על פני היעדים השונים באירופה.
בד"כ מדובר ב"אגרקסקו" על עשרות מוצרים שונים ועל מאות יעדים סופיים, אליהם מגיעים מוצרים אלו.
לצורך יישום המודל לא נחטא לאמת אם נאמר שניתן לחלק את מגוון המוצרים המיוצאים לשלוש קבוצות עקריות, השונות האחת מהשניה במהות המודל:

3.1.1 מוצרי הפרחים למיניהם, שיחס נפח/משקל שלהם קרוב מאוד וקיימת לגביהם מגבלה, הם חייבים להישלח באוויר, בגלל אורך חיי מוצר קצרים מאוד.

3.1.2 ירקות ופירות מסויימים שגם בהם היחס נפח/משקל קרוב וגם הם חייבים להישלח באויר בגלל אורך חיי מוצר קצרים לדוגמא: תות-שדה, מלונים וכד'.

3.1.3 סוגי תוצרת אחרים, ירקות ופירות שנשלחים בד"כ בים אך האלטרנטיבה שלהם היא לצאת באויר, כאשר יש צורך למלא פנירון לא מנוצל במטוסים.

חלוקה זו נבחרה כיוון שגם תעריפי ההובלה אחידים בתוך כל קבוצה ושונים בין הקבוצות.
עד כאן ציפינו שהמודל יתן לנו את היחס שכל קבוצה תהווה מבחינת המשקל או הנפח בטיסות או בהפלגות של אותו שבוע ליעדי השטעון והיעדים הסופיים.

3.2 אנו הזדמנו למודל אינפורמציה על סוגי כלי התובלה השונים הן בים והן באויר. על מנת לא להעמיס יתר על המידה בחרנו באלטרנטיבות של שלושה סוגי מטוסים, והקומבינציות ביניהם:

3.2.1 מטוסי ג'אמבו 8747, מטוסי 8707, כמטוסים הכורים, וטיסות קו המתקיימות מידי יום לאירופה.

3.2.2 בחרנו בשלושה סוגי אוניות שונים.

3.2.3 בתוך אירופה בחרנו בשלושה סוגי כלי תובלה, משאיות בנפח של 60 מ³, קרונות רכבת בנפח דומה ושרותי התובלה האוירית הפנים אירופיים, עבור אותם סוגי תוצרת שחייבים להמשיך באויר ומרחק תובלה זה מחייב מעבר בין שתי מדינות, מנקודת השטעון ועד ליעד הסופי.

כאן ציפינו שהמודל יבחר עבור הפתרון האופטימאלי את קומבינצית התובלה הזולה ביותר, מבחינת סוגי הכלים שעלינו להשתמש בהם לשבוע הנדון.

3.3 מבחינת יעדים חילקנו את אירופה ל 10 אזורים ראשיים כאשר חלוקה זו די תואמת את החלוקה הגיאוגראפית בין המדינות ואלו הן: -

3.3.1 איטליה, צרפת, אנגליה, צפון גרמניה ודנמארק, מרכז גרמניה, דרום גרמניה, ארצות השפלה, אוסטריה, שוייץ, סקנדינביה. בכל אזור כזה נבחרה עיר מרכזית ליצג את האזור.

3.3.2 מתוך צפיה והכרת הנתונים בחרנו במספר מרכזי שטעון אלטרנטביים והבחירה היתה בהתאם לאפשרויות התובלה מהם לכל האזורים האחרים הן ביבשה והן באויר.

האלטרנטיבות שנבחרו לבדיקה היו פרנקפורט בגרמניה, אמסטרדם בארצות השפלה, פריס בצרפת, לונדון כמרכז קשר לצפון אירופה סטוקהולם לסקנדינביה, ציריך ורומא לדרום ומרכז אירופה. כנקודות שטעון ימיות נבחרו, מרסיי, טריאסט לדרום ומרכז אירופה, רוטרדאם, לונדון, הליסנבורג לצפון אירופה.

3.3.3 הנקודה המנחה היתה שמחיר התובלה, מכל מרכז שטעון כזה לנקו' השונות בתוך האיזורים הנזכרים ב- 3.3.1, אחיד פחות או יותר. כאן ציפינו שהמודל יבחר את צמתי השטעון הטובים ביותר להעברת התוצרת לאזורים השונים, המהווים יעדים סופיים.

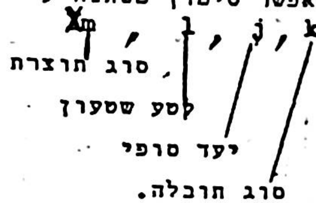
כל הפרטים הנ"ל הצטרפו יחד למאמץ כביר של איסוף מאות פרטי אינפורמציה להזנת המודל.

נקבעה מערכת קידוד אחידה שנתנה זיהוי חד ערכי ומשמעותי למשתנים, כיוון שהבעיה היגיעה למימדים גדולים.

3.4 שיטת הסימון.

כאמור השתמשנו בחבילות התוכנה M.P.S.X. של י.ב.מ.

ה- M.P.S.X. מאפשר סימון משתנה ע"י שמונה פוזיציות.



3.4.1 סוג תוצרת.

כאמור חלקנו את התוצרת לשלושה סוגים עיקריים.

B = תות-שדה וירקות ופירות שחייבים להוביל באויר.

F = פרחים למיניהם.

Y = ירקות פירות ואחרים שבדרך כלל מיוצאים בים.

3.4.2 קטע שטעון יסומן בשתי האותיות הראשונות של קצות הקטע כגון:

לוד-פרנקפורט יסומן IF.

אשדוד-מרסיי יסומן AM.

3.4.3 סוג תובלה.

סוגי התחבורה הראשיים הם:

A - יסמן תובלה אוירית.

S - יסמן תובלה ימית.

T - יסמן תובלה יבשתית ברכבת.

V - יסמן תובלה יבשתית במשאירת.

3.4.4.1 באויר אנו מבחינים בשלושה תת-סוגי תובלה.

AJ - יסמן מטוסי ג'אמבו B 747

AB - יסמן מטוסי B 707

A - יסמן טיסות קו באירופה או מן הארץ לאירופה.

3.4.4.2 גם בים אנו מבחינים בשלושה תת-סוגי תובלה.

S1 - אניה מסוג 1.

S2 - אניה מסוג 2.

S3 - אניה מסוג 3.

3.4.4.3 דוגמאות: -

א. סימון של משלוח פרחים לשטוקהולם דרך פרנקפורט כיעד לשטען אוירי, המשלוח יצא במטוס ג'אמבו מן הארץ.



ב. משלוח ירקות לפאריס דרך מרסיי באניה מסוג 1 .



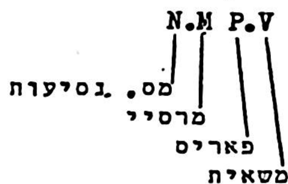
ב1. המשך של אותו משלוח ממרסיי לפאריס במשאית .



ג. מס' הטיסות שיבצע מטוס ג'אמבו בקטע לוד-פרנקפורט .



ד. מס' נסיעות של משאית בקטע מרסיי פאריס .



וכך הלאה סימנו את כל הקטעים האפשריים .

3.5 בנוסף לכל האינפורמציה הנ"ל הזרמנו למודל את RHS הביקוש מכל קבוצת מוצרים בכל יעד בשבוע הנדון בטונות .
כמובן שלפעם הראשונה היה צורך לקדד את כל האינפורמציה הנ"ל על מנת ליצור קובץ מתאים שיכיל את כל הבעיה.מכאן והלאה כל העדכונים נעשה בשיטה ממוכנת .

3.6 נכתבה תכנית בפורטרן שמתפקידה הוא לבנות רשומות במבנה המוכר ע"י ה- M.P.S.X מכרכיסי פרמטר . תכנית זו יוצרת קובץ עדכונים מתאים ופקודות M.P.S.X מתאימות לעדכון והללו מעדכנות את הבעיה הקיימת על הקובץ .

3.6.1 כאמור לעיל הבעיה בנויה להיקף של שבוע יצוא אחד ועל מנת להקיף את העונה כולה יש צורך ליצור RHS עבור כל שבוע משבועות העונה, ולהזרימו פעם אחר פעם, על מנת להפעיל עליו את המודל.

3.6.2 אינפורמציה נוספת לעדכון היא שינויים שחלים בפרמטרים של התכנית, במחירי הובלה, באפשרויות שטעון או פנירון אוירי, בסוגי מטוסים או אוניות אותם רוצים לבדוק. גם שינויים כאלו נבנים אוטומאטית באמצעות תכנית בפורטרן שיוצרת את קובץ העדכונים עבור המודל.

4. תוצאות ולקחים.

גודל הבעיה במודל שלנו הוא כדלקמן: -

942 שורות, 1958 משתנים, כאשר 117 מהם הם משתנים שלמים (Integer Variables).

הכמויות שהוזרמו למודל היו בטונות לשבוע, על מנת להמחיש את 'מריחת' הכמויות על פני השבוע כולו ובהנחה שהכמויות מפולגות אחיד על פני השבוע, הכנסנו מגבלה נוספת למודל שקבעה: -

בכל טיסה שיוצאת ליעד מסויים תצא לא יותר מרבע הכמות השבועית, מגבלה זו 'מרחה' לנו את הכמויות על פני השבוע.

פתרון אופטימאלי ראשון, התקבל לבעיה הנ"ל אחרי 6 דקות. וכאן החל ה- M.I.P לבצע branch and bound ומצא ארבעה פתרונות פיזיביליים בודדים כאשר הפתרון האופטימלי הבדיד התקבל לאחר 65 דקות ריצה.

לשבוע מסויים עם כמות כוללת של 2732 טון תוצרת כאשר מתוכם 1570 טון בים, המודל ממליץ על בצוע 37 טיסות חכורות לאירופה ולצורך זה להשתמש בחמישה מטוסים ועל שתי הפלגות בים.

המודל מצביע על יעדי הנחיתה העיקריים של המטוסים החכורים, ועל היעדים אליהם יש להטיס תוצרת במטוסי קו.

המודל ממליץ על בצוע נחיתות כפולות בחלק מן הטיסות החכורות.
(נחיתה כפולה היא נחיתה בשני יעדים במהלך מסלול אחד), ועל הכמויות שיש להציא לכל יעד בטיסות אלו.

המודל ממליץ על שיטת ההובלה בתוך אירופה, מספר משאיות או קרונות רכבת, וכן טיסות הקו הפנים אירופיות שיש להשתמש בהן על מנת להעביר את התוצרת ממרכזי השטעון ליעדים הסופיים.

4.1 תוצאות מעניינות שהמודל ממליץ עליהן: -

4.1.1 העברת תוצרת מן הים לאויר לטיסות החכורות, על מנת למלא מקום פנוי במטוסים החכורים, וכן היעדים שעבורם יש לעשות כן.

4.1.2 המודל הצביע על בצוע הפלגה לאנגליה על מנת לספק בקוש לתוצרת שמהלכה בים לצפון ומערב אירופה.
התוצאה נבדקה ע"י אנשי התובלה ונמצאה סבירה לבצוע.

כיום על מנת לקבל תמונה של כל העונה, שבוע אחרי שבוע, אנו מריצים במחשב כל שבוע בנפרד בלי תלות בין השבועות, וכך אנו מקבלים את כל התוצאות הדרושות ללא תלות בין השבועות, כאן נכנס האדם לפעולה ומכניס את גורם התלות לתוצאות כך שתתקבל התפלגות הגיונית שהחברה יכולה לעמוד בה בחוזי שכירות.
לדוגמא: אם התקבלה תוצאה במודל שבשבוע מסויים אנו זקוקים ל- 5 מטוסים ובשבוע שלאחריו לארבעה ואחריו שוב לחמישה, אזי ההחלטה היא לחכור לשלושת השבועות הנ"ל 5 מטוסים, בידיעה מלאה שחוטאים כאן לאופטימום וכך הלאה.
כיוון שהתפלגות הכמויות על פני העונה היא בקרוב נורמאלית אזי גם התפלגות צריכת כלי התובלה מתקרבת לאותה התפלגות.

5. כוונות הפתוח.

בקטע זה האפשרויות הן רבות.

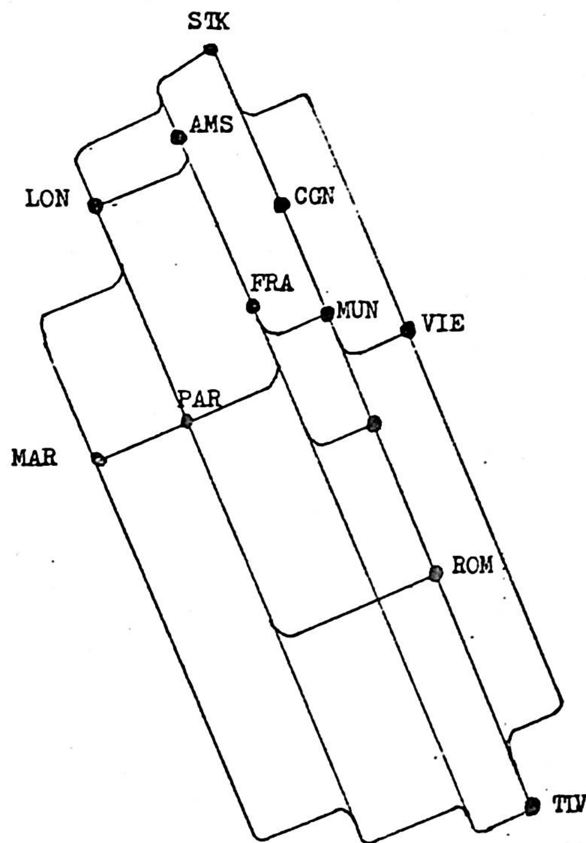
5.1 בכוונתו להכניס למודל את מומנט התלות בין השבועות, ולהפוך את העונה כולה ליחידה אחת עם התפלגות המתאימה בין השבועות ובתוך השבועות.

5.2 להביא לאינטגרציה בין הקטע המקצה אופטימאלית בין השווקים השונים על סמך אינפורמציה מאותם שווקים, לקטע המקצה את אמצעי התובלה להובלת ההקצאה האופטימאלית הנ"ל במחיר מינימאלי.

5.3 ובשלב אחרון לנסות להפוך את המהלכים הנ"ל לאנטראקטיביים בין המשתמשים בפועל, קרי אנשי תכנון באגף התובלה והמחשב, כאשר התכנון הפרילימינארי ייעשה ב-batch, אך כאשר תתקבל ההחלטה אלו כלי תובלה לחכור, והתפעול יעבור למדיום השבועי ביצועי להכין כלי אנטראקטיבי שיקצה את התוצרת בצורה אופטימאלית אך יתחשב גם במגבלות אמצעי התובלה שנחכרו.

6. סכום.

כוונת מאמר זה הייתה להציג כיצד ניתן להשתמש בתכנון מתימטי לפתרון בעיות מעשיות בחברת שווק גדולה. יש לזכור דבר חשוב המהווה תמיד נר לרגלנו: - הפתרון הניתן ע"י מודלים מתימטיים הוא בבחינת הצעה למשתמש התחלה טובה יותר לקבלת ההחלטה מכל נקודת התחלה אחרת שיכול להגיע אליה המחליט ויהיה המהיר והשנון ביותר. וזאת בגלל ריבוי הנתונים וקיום הקומבינציות ופרמוטציות רבות של פתרון. אין בפתרון של המודל בבחינת "כזה ראה וקדש", כאן בא מקומו של המתכנן, מקבל ההחלטה, להכריע ותמיד אחריות ההחלטה תהיה עליו.


STK
AMS
LON
CGN
FRA
MUN
VIE
PAR
MAR
ROM
TLV

AN ON-LINE CONTAINER CONTROL SYSTEM FOR ISRAELI PORTS

Hadar Matmon, Israel Ports Authority

Abstract:

An On-Line Control System has been designed and implemented by the Israel Ports Authority for the control and analysis of container traffic in the ports of Ashdod and Haifa. The system provides facilities for reporting on the arrival, in-port-handling and departure of containers, both import and export. At each and any stage of its career, the location of a container and any of its status fields (e.g., contents, ship/port code, owner, etc.) may be monitored, and, if desired, updated.

The system software consists of eighteen user-addressable modules together with over 40 transparent subroutines for I/O handling, user communication, logical checking, conversions, etc.

The system provides management reports on the status of the storage facilities, stuffing/unstuffing information, embarking/debarking lists and empty stock by specified criteria.

The IPA Container Control System is written in PL/1 and implemented on the IBM 370/168 of the Office Mechanization Center in Jerusalem.

Each facility of the system is an independent load module and can be executed either on-line under TSO or as a background job in batch. File access may be simultaneous by any number of users for query/update purposes. Maximum storage requirements for the most complex function is 320 kilobytes plus one dedicated 3330 disk pack and drive.

The system utilizes advanced technologies in multitasking, screen formatting (on-line terminals are alphameric CRT display

units), and file organization: the major data sets in the system are VSAM allowing for maximum efficiency in direct-access query/update and sequential report generation as well as optimum disk utilization.

Historical files of departed containers are maintained for purposes of billing and statistical analyses.

Background:

With the increasing trend toward unitization and containerization in maritime shipping, the ports of Israel, mainly Ashdod and Haifa, are faced with the technological and informational challenge of managing the steady and dependable flow and storage of approximately 5000 containers per month import and export. These containers, in themselves a significant investment to the shipping companies which own them, must be available for speedy dispatch when the delivery van calls at the port and when exported, placed in such a manner so as not to cause unnecessary delays to the extremely expensive handling equipment (transtainers, gantries, trolleys, etc.).

World-wide use of this method of goods shipment has led to a measure of welcome standardization, namely: uniform sizes and types of containers; regular ports of call for particular containers; and a universal identification code scheme. These factors, together with the difficulty of tracking, by hand, individual units in vast crowded storage sites, have naturally led to the development of computer programs and data bases devoted to this application. A beneficial side effect has been the improvement in accounting speed and accuracy. Some of the more

prominent world ports, where such programs are already in daily use, include New York, Liverpool and Bremen.

Real-world procedures:

1. Export

   a. Full containers arriving at the port by truck or rail are accompanied by a storage document listing all the pertinent data of the container, such as gross weight, ship name and destination port. The container is placed in a storage location allocated to that ship and port, with further subdivisions of weight class and, if necessary, facilities for powering the cooling units of refrigerator containers holding perishable goods. At a specified time, all containers bound for the same ship are transferred in sequence to pier side where they are on-loaded.

   b. Exporters who do not have a sufficient quantity of goods to justify an entire container, stuff, using port facilities, together with other exporters in the same class having consignments bound for the same port. Upon exit from the stuffing shed, treatment is the same as for full export.

   c. Due to Israel's negative balance of trade, more full containers are imported than exported. Since a ship always sails with the same number of containers, the difference is made up by shipping empty containers belonging to the same shipping company and also to the route being sailed by the ship.

2. Import

Incoming ships are preceded by a manifest supplied by the shipping company detailing all the expected containers and their attributes. The delay between manifest and ship arrival allows for a leisurely and careful preparation of an input file for the creation of basic container records. This creation can be performed at convenience in an economical bath run.

a. Door-to-door containers are destined for direct delivery to importers without intermediate processing.

b. Pier-to-pier containers include consignments for more than one importer. These containers are un-stuffed in the port facilities and the now empty container is returned to stock.

c. Empties (see paragraph 1.c).

Import containers may be claimed by importers on presentation of a gate pass from the shipping company upon entry to the port or a delivery order in the case of empty containers. Pier-to-pier containers and empty containers required for stuffing are transferred to the shed when requested by the shed foreman.

In all cases where full containers are required, the container is specified by identity number and the computer immedidately reports back its current location. In all cases where an empty container is required, specific identity is immaterial and the computer presents a list of suitable candidates, given the criteria of size, type, company and sailing route for the desired container

together with their current locations.

Working files:

There are four main data sets associated with the system:

1. the current container file
2. the map file
3. the transient departed containers file
4. the daily transaction log file

The current container file has a record for each container currently in the port or expected on import. Each record lists the identity of the container and its physical characteristics as well as data referring to its present "incarnation" in the port (arrival and departure dates, ship and port names, etc.). Each site at which the container has been stored, including the current one, is reflected in a record trailer so that the container's entire career in the port can be audited.

The transient departed file contains a copy from the current file of each container that has left the port together with a computer time stamp showing the time of report of transaction. The record in the current file is deleted. The transient file is maintained for about two weeks for reporting purposes. It is then transferred to a generation historical tape data set.

The daily log file contains a copy of each container record which has been created, amended or deleted and the computer time stamp of report. This file is cleared daily after serving any required audit purposes or recovery in case of current file crash.

The above three data sets must simultaneously serve the requirements of concurrent update from multiple terminals and batch, and sources for report generation. PL/1 optimizer provides data

set integrity between concurrent jobs only for direct update files (with keys). Since the record keys are arbitrary, indexed organization was decided on for these data sets; an organization amenable also to economical sequential processing. However, due to the highly dynamic nature of these data sets, namely a very high rate of record addition, ISAM with its overflow disadvantages was discarded in favor of VSAM.

The map file presents a snapshot of the port storage area as a three-dimensional array wherein each storage slot is designated by Cartesian coordinates. The elements of the array contain the allocated attributes for the slot determined beforehand by the service management and the slot's current contents, be it a container identifier, the "forbidden" designator or blanks signifying availability. Since the physical records of this file are fixed in number and in length and are identifiable by natural numbers is strict sequence, the PL/1 Regional (1) data organization was decided upon as most suitable.

User Communication:

When used on-line, the system provides facilities for terminal I/O in a form easily learned by non-technical personnel. Utilizing a generalized communications package developed by the Israel Ports Authority Haifa computer department, and compatible with Pl/1 and COBOL, the user is presented on screen with a stylized form for each application on which blank spaces need only to be filled in. Input interpretation is performed automatically and restore-screen facilities are included in case of hardware failure or inadvertent clearing.

Up to eight transactions can be recorded in a single press of the "ENTER" key. When only individual queries are expected, simple single displays and responses are utilized with programmed prompting and using normal PL/1 TSO I/O.

Each application program is considered activated on-line by default. If batch processing is required, an EXEC parameter is usually provided to specify this.

Application Programs:

Application programs are divided into two classes: file up-dating programs and query/report generating programs.

Five file updating programs are in normal use for the current container file:

1. KLITA - creates new basic records for incoming containers, whether import or export. Accepts data concerning physical characteristics and current "incarnation" and performs logical checks.
2. ASHLAMA - accepts information concerning actual arrival of container and placement in storage. Updates container record with arrival data and builds first location trailer. If container is unrecognized, activates KLITA as subroutine. Updates map file.
3. HAZAZA - accepts data concerning in-port movements of container as result of shifting or moves to or from shed or repair. Adds new trailer and updates map file.
4. IDKUN - accepts report of container exit from port. Adds exit date, spills record to transient file after

deleting from current file. Updates map file.

5. MISHALA - activated from gate to determine suitable location for incoming export container. If containe is unrecognized, activates KLITA as subroutine.

One program, HAKZAA, is used by the service manager to modify the elements of the map file for some arbitrary rectangular subarray with respect to the allocated attributes used in recommending storage sites for incoming containers given their characteristics.

A general amendment program, TIKUN, can be used by qualified personnel to survey all fields of any record in the current container and map files and enter arbitrary changes in said record.

All I/O accesses to the files are performed by tailored plug-in modules allowing extensive changes in file organization without requiring recompilation of the main programs.

All normal file updates are filtered through a general service subroutine MEZIZ thus making each main program device independent.

The query/report generation programs are as follows:

1. MAPA- a snapshot of the present status of the map file.
2. CARP- print unloading cards for containers arriving on a specified ship/trip.
3. CART- print loading cards for containers due to be loaded on specified ship/trip.
4. DOCP- print list of all containers due to arrive or actually unloaded for specified ship/trip.

5. DOCT- print list of all containers actually on-loaded for specified ship/trip.

6. ITUR- find specified container.

7. EMPT- find empty candidate containers by criteria.

8. PIZUL- list all containers due to be unstuffed for specified ship/trip.

9. STOCK- list all containers currently in port.

10. GIL- list age of all door-to-door import containers.

11. EXP- list all containers due to be exported on specified ship/trip in order of ports of call.

All reports are appropriately sorted using standard I.B.M. sort packages activated in line from the PL/1 application program.

Special TSO procedures:

Special Logon procedures have been devised by the O.M.C. systems staff for use by this system. Required files are automatically allocated together with a tailor-made command library so that users have only to type the first three or four letters of the desired program name in order to activate it.

Since the user may request a report program including a sort routine requiring a large core, standard region size for this system has been set at 320 kilobytes.

The current container and map files are dumped daily so that in case of disk failure, recovery is relatively simple.

Future Developments:

Two future developments of far-reaching significance are now undergoing preliminary study:

1. A feasibility study is being conducted to determine whether hand-held wireless terminals can be employed by field supervisors to report instantaneously or movements by containers into, out of and around the storage area.

2. The Ports Authority, in conjunction with a manufacturer of container handling equipment, are considering the fitting of cranes with microprocessors and guidance equipment. This would allow the central computer to issue direct orders to the crane to move to a required site. The crane continuously would feed back its position in three dimensions.

קבוצת דיון מס' 8:

"רשתות מחשבים"

יו"ר: ד"ר י. פז

י. פז

ל. פפר

מ. להב

ON PATH-BY-PATH AND NODE - BY - NODE ROUTING IN LOADED COMPUTER-NETWORKS

by
Israel Paz *

ABSTRACT

The capability, P, of a loaded computer network to handle additional traffic demands, has been shown to depend on the homogeneity, H, by which the existing traffic load is distributed through the network.

This result is used here, for developing two routing algorithms which care for the best value of P at each time sample. They do so indirectly by ensuring that the changes in H, remain below a certain value: h, when new traffic increments are accomodated. The value of the oonstant, h, is related to the amount of computational burden one wishes to assign to the optimization effort, and should be obtained on the basis of previous experience with the network and its environment.

1. INTRODUCTION

The routing optimization problem, has been known for sime time [1,2,3] to present an interesting but difficult challenge that no conventional optimization tools could efficiently meet. Heuristic sumoptimal routing strategies have therefore been developed [2,4,5] and some even verified by computer simulations [6].

For circuit switched networks we have worked out, in a previous paper [3], some relationships between the way in which the present traffic load is distributed through a network and its capability to handle future, additional demands. In this paper we describe two suboptimal routing algorithms based on those relationships. The reader is refered to Ref. 3, for notations and definitions used in the sequel, some of which are repeated (with minor changes) in the next paragraph for completeness.

2. THE CAPABILITY OF A LOADED COMMUNICATION NETWORK

Let $i_j = \frac{x_j}{c_j} \triangleq$ the flow through the branch b_j in the network, as normalized against its capacity c_j. (1)

and let

$\mu = \frac{1}{m} \sum_{b_j \in B} i_j \triangleq$ the "loading" on the network (2)

with m the number of branches contained in its branch-set B

Let also $H = \sum_{b_j \in B} (\mu - i_j)^2$ (3)

define the homogeneity by which the traffic load in the network is distributed through it (with lower values of H representing more homogeneous flow distributions).

The capability of a network to handle future traffic demand increments (or shortly its "capability") is then defined [3] as

$$P = \sum_{b_j \in B} w_j (1 - i_j^2) \quad (4)$$

* Dr. I. Paz is with the Ministry of Defence - Israel.

with, w_j -the (simplified) operational "importance"[2] of the branch b_j as defined in Ref. 3.

The capability P so defined has been related in Ref. 3 to the homogeneity, H, of the flow distribution through the network. It has been indicated there, that P tends to increase with decreasing values of H if the loading, μ on the network can be kept constant.

This relationship between P and H (as given in Ref. 3) could however not serve the actual routing process, because P, as defined, is an a-posteriori indicator, that can only be evaluated after the routing is already decided upon.

To overcome this difficulty and obtain a routing strategy that maximizes P while loading the network with the required sequential traffic demand increments, we will operate on H. Namely, we suggest algorithms that will accomodate the traffic increments through the network, by keeping the increments on μ as low as possible (shortest possible routes) while bringing H to its minimum, possible values.

3. ASSUMPTIONS AND DEFINITIONS

The network, N, and the initial flow distribution through it, x, for a certain loading μ being given, the routing-algorithms to be described are based on the following set of assumptions.

1) Feasibility The (previous) network loading is such that the expected additional traffic demand increments can be accommodated by the network, without saturating any of its branches.

2) Uniform distribution of demands. All terminal-pairs of the network have equal probability of being requested to handle additional traffic increments.

3) The Demand-Increments. The Demand-Increments are obtained as follows:

   a) The demand is sampled sequentially in time and the traffic requests at each given time sample are assembled in a Demand-Vector D as described in Ref. (1)**

   b) Each Demand-Vector so obtained is substracted from the previous one (element by element) and the resulting differences (ΔQ^k) are entered in a "Demand-Increments-Vector" (ΔD) in corresponding order.

4) No Rerouting. New traffic increments are routed through the loaded network without rerouting flows that already exist there. We recall that this requirement was one of the difficulties in solving the optimal routing problem in real time* because it called for a new solution to the overall flow-problem at each time sample. Clearly, however, only suboptimal results can this way be expected.

---

* As a by-produce (in telephone network applications) this assumption should presumably also lead to simpler instrumentation needs.

** Tentatively a requirement matrix $R=(r^{\omega,\delta})$ can also be used[2].

5) No rerouting while "releasing calls". The negative entries in ΔD correspond to flow decrements. These flow decrements are effected in the network, without changing flows in branches that do not belong to the pertinent "released paths". In practical (e.g. telephone) applications, such decrements occur when the number of calls requesting a termination of service (a "release") is larger than the number requesting a new interconnection, between the considered terminal pair. We also assume here that the network has been cleared of the decrements mentioned above, before the optimal routing algorithm, to be used, starts.

6) Positive demand increments. Only the positive (and non zero) remaining, entries of the vector ΔD are therefore taken in consideration in the proposed routing optimization rules.

7) Smallness of Increments. This assumption can be justified (in telephone applications) as follows:

- It can be assumed that in a loaded network and with high sampling rates, some kind of statistical equilibrium exists. Presumably the number of calls that request service (at a given terminal-pair and given time-sample) approximately equal the number of those asking for a "release".

- Most of the paths about to be released can, therefore, be reused by the new requests for service, only the remaining ones having to be actually routed a-new and only they will therefore be considered "demand-increments".

We thus require here that:

$$\left.\begin{array}{l} \Delta x_j \ll c_j \\ c_j \gg 1 \end{array}\right\} \forall b_j \varepsilon B \qquad (5)$$

with $\Delta x_j = \Delta x_j{}^k$ being the flow increment in branch b_j for the kth request. (Superscripts will be omitted whenever possible.)

8) Independent Handling of Increments. Each entry in ΔD is handled separately. At each time-sample we scan the whole vector and perform routing for each terminal pair independently.

4. PATH-BY-PATH AND NODE-BY-NODE ROUTING

The eigth assumption made above, will actually lead to a sequential selection of sub-optimal paths. The required additional traffic flow at each terminal-pair being handled independently at each given time sample.

Considering now the way in which the switching necessary, for assembling the chosen paths, is technically performed we may have:

a. Routing under central control

b. Routing under local control.

The routing under central control needs a "centralized" network. The paths being selected sequentially but implemented as a unit (proper switches operated at intermediary nodes simultaneously) under central control. We shall call this type of routing Path-by-Path Routing.

In distributed networks, the possibility arises to choose and assemble a path, under local-control. We still handle one requirement (terminal-pair) at a time. However, we perform then a so called "Node-by-Node routing". In this kind of routing, each node, obviously, should have the necessary amount of "intelligence" needed for making a local decision and implementing it. Such a decision, has to be made for each and every path that reaches the considered node during its (propagating) assembly process. It should select, among the branches incident with this node, the one that's "best" for carrying on the path-assembly process.

In what follows we present two algorithms. One for choosing the best path in a path-by-path routing, the other for selecting the "best" branch that shall join a path next, in the node-by-node routing.

5. PATH-BY-PATH ROUTING

In this method, a "Central Controller" calculates for each positive non-zero ΔQ^k, the best available path between the pertinent terminal pair, k.

The results previously obtained indicate that small values of H lead to an improved "Capability" of the network. In the routing algorithm described below, the best path is therefore considered to be the one which gives the minimal ΔH_s when the flow increment is accommodated by the network, ΔH_s being the change in H brought about by accommodating ΔQ^k through the branches that form path 1_s^k.

To simplify matters, we make the following assumption in addition to those already made above. We call this new assumption single path routing, and it requests that the entire flow needed for satisfying an increment of demand shall be handled by a single-path, i.e.:

Let $P_{1,k} \triangleq \{ 1_p{}^k \}$; $p=1,2\ldots\ldots P_k$

be the set of all paths of length 1 available between terminal pair k, and $1_s{}^k$ be the one that has been selected. Our new assumption is equivalent to requesting that

$$\Delta x_j{}^k \equiv \begin{cases} \Delta Q^k, & \forall b_j \varepsilon 1_s{}^k \\ \text{zero if } b_j \not\varepsilon 1_s{}^k \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

5.1 The Algorithm

The proposed routing procedure is now as follows

Step 1 Initialization

Step 2 Test if terminal-pair k needs service, i.e. if $\Delta Q^k > 0$

Step 3 Find (E.g. by using the method indicated in Ref.8) $P_{1,k} = \{1_p{}^k\}$, of all paths with length 1 interconnecting terminal-pair k. Let p_k be the number of its elements. If the set is empty for the chosen 1, increment 1 by one and go to step 8, otherwise proceed.

Step 4 Test if $1_p{}^k \varepsilon L_{1,k}$ satisfies:
a) the single-path-routing assumption as given in equations (5)
b) feasibility as given below

$$\Delta i_j + i_j \leq 1; \ \forall b_j \varepsilon \ 1_p{}^k \ \ldots\ldots\ldots \qquad (7)$$

If both tests are successful, set p=α and proceed to next step, otherwise increment p to p+1 and go to step 7

Step 5 Calculate
$$\Delta H_\alpha = \sum_{b_j \epsilon \ell_\alpha^k} \{ \frac{1}{c_j{}^2 \Delta x_j} [2x_j + \Delta x_j - 2\mu c_j] \} \qquad (8)$$

Step 6 Test if $\Delta H\alpha < h$ .... (9)
with h a small positive constant, to be determined on the basis of previous experience. If the test gives an affirmative answer, select path α as the optimal one (i.e. set α=s and $l_\alpha^k = l_s{}^k$) assemble it and jump to step 9. Otherwise set p=α+1 and proceed.

Step 7. Test if $p \leq p_k$
In the affirmative, return to step 4. Otherwise increment l to l+1 and proceed.

Step 8 Test if $l < l_{max}$
with l_{max}, obtained by following the considerations given in par. 5.2 below. In the affirmative, reset p and return to Step 3.
Otherwise reject ΔQ^k increment k by one, update the grade of service indicator (and issue an alarm if his value has passed a predetermined threshold value).

Step 9 Let M be the highest numbered terminal pair requesting service.
(i.e. M = max. k, $\ni \Delta Q^k > 0$)
Test if k < M.
In the affirmative, return to Step 3, to restart the algorithm for the next terminal pair to be serviced. Otherwise Stop, reset all running indexes and wait for the next time sample.

Remarks:

1. The equations used in this algorithm and the considerations that justify its rules are omitted here for brevity.
2. The alarm issued during Step 8 indicates that the grade of service has shifted out of its permitted region of existence. This shift may indeed occur due to the sub-optimality of this rotuing method. The alarm will therefore serve to indicate when does an accurate solution to the optimal routing problem need to be obtained again.

5.2 Discussion on the Path-by-Path Routing

The value of ΔH used in the algorithm is calculated elsewhere.[11] Clearly, the maximal improvement in the homogeneity of flow is obtained if, for each terminal-pair, the path providing the minimal value of ΔH is selected. In the proposed algorithm, however, we selected, instead, the shortest path for which $\Delta H_s < h$ holds. As the paths are obtained sequentially, in order of increasing length, this should lead to savings in the computional effort.

The value of h is to be determined experimentally and depends on the network configuration, the amount of computation one can tolerate at the Central-Controller for each path selection and, in telephone networks, on the permitted Grade-of-Service as well.

The value of ℓ_{max} used in the algorithm (step 8) can be predetermined in several ways, e.g.:

a. Assuming Δx_j max to be the maximal expected or permitted demand-increment, we may then use for each k, the following definition

$$\Delta\mu \triangleq \frac{1}{m} \sum_{b_j \epsilon \ell_s^k} \left(\frac{\Delta x_j}{C_j}\right) = \frac{1}{m} \sum_{b_j} \Delta i_j \quad (10)$$

from this definition a bound on $\Delta\mu$ has been calculated to be

$$\Delta\mu max = \frac{1}{m} \sum_{b_j \epsilon l_{max}} \frac{\Delta x_{jmax}}{C_{j\ min}} \leqq 2\mu \quad (11)$$

with $l_{max} = \max_{k,p_i} l^k_{pi}$

which gives $lmax \leqq \dfrac{m \cdot 2\mu \cdot C_{jmin}}{\Delta x_{jmax}}$ (12)

b. Relating imax to the diameter (K) of the network [9], by requesting that

$$lmax \leqq K + r$$

with r a positive integer.
In this case r could e.g., be determined on the basis of previous experience.

6. THE NODE BY NODE ROUTING

In distributed networks [7], each node is assumed to have the instrumentation necessary for making routing decisions on a local basis. A path is then assembled sequentially, branch by branch. The decision as to which branch is to join the path next, is taken at the end-node of the last branch already in it. This kind of routing is known as node-by-node routing. For the network to perform it, each node:

a) has to know, for any k, which of the branches connected to it belong to some of the sets $\{l_p^k\}$ with l=1,2.....lmax, and p=1,2....P_k.

b) shall know a simple routing rule by which to choose the best branch that has to join the path next.

The results obtained previously, lead to the following algorithm for our sub-optimal node-by-node routing.

6.1 The Node-by-Node Routing Algorithm

Step 1 Of all the branches incident at the considered node, select those belonging to the sets $\{l_p^k\}$, with l=1 .... lmax. (Start with the first branch in the set of minimal length l).

Step 2 Out of the branches so selected, retain only those that can accommodate Δx_j, i.e. those where:

$$(i_j + \frac{\Delta i_j}{2}) - \mu \triangleq \nu_j < 0 \ .... \quad (14)$$

Step 3 If there are many such branches, choose the one for which $\left|\frac{\nu_j}{C_j}\right|$ is max.

Step 4 If no such branch exists, i.e. $\nu_b \geqq 0$ for all branches incident at the considered node, then, among those feasible choose the one with minimum value of $\{|\nu_j/C_j|\}$ which is equivalent in practice to choosing the one of largest capacity.

6.2 Discussion of the Node-by-Node Routing Rule

Let equation (8) be rewritten as follows:

$$\Delta H = \sum_{b_j \varepsilon 1_p}^{k} (\Delta H_j) = \frac{2}{\Delta x_j} \sum_{b_j \varepsilon 1_p}^{k} \left(\frac{\nu_j}{c_j} \right) \quad (15)$$

with

$$\nu_j = |_\ i_j + \frac{\Delta i_j}{2} - \mu_|$$

Where Δx_j has been taken out of the summation because it is a constant over all the branches in the path.

Consider at each node, only those branches that belong to paths in $P_{1,k}$ not yet discarded at previous nodes, and able to accommodate the requested flow increment Δx_j.

We remark that in order to minimize ΔH, the values of ΔH_j shall be choosen as follows:

a) Whenever $\nu_b < 0$, the branch that gives the largest value for $\frac{\nu_b}{c_b}$ shall be chosen.

b) If all eligible branches incident at the considered node have $\nu_b \geq 0$, then the best we can do is to choose the branch giving the smallest local contribution to ΔH, i.e. smallest value of $\left|\frac{\nu_j}{c_j}\right|$.

The above remarks can actually serve as a proof of the algorithm.
Additional insight into it can be obtained as follows:

Let $\hat{i}_j \triangleq i_j + \frac{\Delta i_j}{2} \triangleq$ the "incremented branch flow" among branches with same ν_j:

If $\nu_j < 0$ (i.e. $\hat{i}_j < \mu$) choose the branch of smallest capacity for continuing the path.

If $\nu_b \geq 0$, i.e. the incremented branch flow at least equals the loading ($\hat{i}_b \geq \mu$), the branch of largest capacity should be chosen.

7. CONCLUSIONS

Two algorithms for routing traffic demand increments through a loaded computer (communication) network have been proposed. The routing deals with one terminal-pair at a time. Its aim is to insure, at each sample in time, that the capability P of the network to handle additional, future, traffic demands increments is kept at its maximum possible value. The algorthims use a relationship that has been proven (heuristically to some extent) elsewhere, which relates P to the homogeneity, H, of the flow distribution existing in the network. The main ideas behind those algorithms have been tested in a slightly different way by computer simulations. The results obtained indicated significant improvements in the grade of service of the network on which they have been applied.

8. ACKNOWLEDGEMENTS

The author wishes to thank Prof. I. Cederbaum from the Technion Haifa for constructive criticism, Dr. N. Shein, formerly with the Tel Aviv University, for interesting comments and editorial help, and Dr. A. Boneh for reviewing the final draft.

9. REFERENCES

[1] Paz I. "Optimal Routing and Optimal Redesign in Communication Nets", Proc. of the 7th IEEE Convention (held in Israel, April 1971)

[2] Cederbaum, I. and Paz. I. "On Optimal Routing Through Communication Nets", Trans. IEEE on Communications, Aug. 1973.

[3] Paz I. "On the Traffic Handling Capability of Loaded Communication Networks" - Proc of the ICC 73 Conf. on Communications held in Seattle, Washington, June 1973.

[4] G. Fultz and L.Kleinrock "Adaptive Routing Techniques for Store and Forwards Computer Communications Networks", 1971 Proc. Int. Conf. on Communications.

[5] Grandjean Ch. "Call Routing Strategies in Telecommunication Networks", Electrical Communications Vol. 42, No. 3 PP 380-391, 1967.

[6] Paz I. and Cederbaum I. "Evaluation of Optimal Routing Policies in Communication Nets by Simulation", Trans. IEEE on Communication, June 1972.

[7] Baran P. "On Distributed Communication Networks", IEEE Trans. Communications Systems, CS-12 1-9, 1964 (see also corresponding RAND Co. Memoranda).

[8] Paz. I. "Path Finding Through Communication Nets", Proc. IEE Vol. 114, No. 7, January 1967.

[9] Elspas B. "Topological Constraints on Interconnection Limited Logic", Proc. of the 1964 Symposium on "Switching Circuits and Logical Design", Vol. 5-164, Oct. 1964, pp. 133-147.

[10] Paz I. "On The Traffic Handling Capability of Communication Networks Homogeneously Loaded", "Networks, 5, 1975, pp. 106-128.

[11] Paz I. "On Path-by-Path and Node-by-Node Routing through Loaded Communication Networks" - in preparation.

A COMPUTER COMMUNICATION NETWORK AT THE ISRAEL POLICE

By

Lawrence Pfeffer*, M. Lahav*, S. Simha[1], G. Redner[1] and M. Allouche[x]

Abstract

A simple communication and computer network of Israeli design and implementation is presented. The Network is based on three computers, namely an IBM System/370 model 145 and two CDC 1700 mini-computers. The IBM system is used for administrative purposes and provides batch and on-line data processing services to a CRT and hard-copy terminal network. The mini-computers form a fully duplexed system dedicated to message switching via a sizable teleprinter network.

The nature of the message switching and the administrative system environments is briefly reviewed. The major portion of this paper is devoted to presentation of the technical aspects of the overall network. Particular emphasis is placed on the processor-to-processor protocol and on the network's virtual terminal support function. The former is the set of ground-rules governing the basic etiquette of system-to-system conversation. The latter hides the specific nature of the CDC 1700 processor and of its numerous tributary stations: the teleprinter network. In closing, the paper describes some of the techniques used in integration testing, as well as some of the difficulties faced by the implementation team.

1. INTRODUCTION

This paper describes a simple computer-communication network, connecting two dissimilar computer systems at the Israel Police National Headquarters in Jerusalem. The Network was planned and designed by ILTAM, acting as the technical advisor to the Israel Police. The system was implemented by ELBIT Computers Ltd., IBM Israel Ltd., and by ILTAM Corporation for Planning and Research Ltd., with the full cooperation of the Communication and Data Processing Divisions of the Israel Police. The Network recently became operational.

The Network is based on three computers, namely an IBM System/370 model 145 processor, and two CDC 1700 minicomputers (see Appendix I). The IBM computer

* - ILTAM Corporation for Planning and Development, Ltd.

1 - ELBIT Computers, Ltd.

x - IBM Israel, Ltd.

(hereafter referred to as the Data Processing Computer: DPC) is used for administrative purposes and provides batch as well as on-line data processing services to a small scale CRT and hard-copy terminal network. The mini-computers, named the Police Message Switching (PMS) System, form a fully duplexed system dedicated to storing and switching telegram traffic via a sizable nationwide teleprinter network. At any time only one of the PMS computers is connected to the IBM System/370.

The network allows message switching within the PMS and simultaneously enables users of the PMS to access the storehouse of application programs and data bases contained in the IBM computer.

2. WHY WAS THIS NETWORK BUILT?

The decision to design and implement this Network has been based on a number of factors each of which has proven to be rational and justified in retrospect. The major decision criteria were the following:

Short Term Economic Benefits:

It was estimated that the capability to access the administrative computer (the DPC) via the message switching system teleprinter network would lead to improved utilization of existing equipment and would prevent procurement of a large number of terminals, modems and communication lines. The estimated long-term savings are in the range of IL 2,000,000 (mostly in foreign currency) and a non-trivial recurring monthly lease charge for communication lines.

Technological Expectations:

The list of expected technical benefits was as follows:

- The ability to utilize existing equipment more effectively and to provide computer access to a large number of police stations,
- An increase in the interaction between two distinct but dependent technological groups within the Israel Police, namely the Communication and Data Processing Divisions.
- An enhancement, because of ELBIT's involvement in the project, of the export potential of Israel's major computer manufacturer.
- Even before inception of this project it was realized that if Israel wishes to be partially self-sufficient in the vital and expensive computer technology, then the government must sponsor applied research

projects in selected branches of the technology. Research projects should ideally meet the following criteria:

I - oriented towards delivery of a needed service or product (that is, not research for its own sake),

II - relatively low risk of failure,

III - relatively inexpensive (with respect to expected benefits),

IV - "road-paver" for future deeper involvement in a branch of technology in which there is national interest.

Looking toward the day when a national pilot computer network will be initiated, the planners realized that there would be an imminent need to acquire in-depth technical know-how in two related technologies: telecommunication and computer-to-computer communication. Consequently it was realized that the Israel Police Network would be both economically justifiable on its own merits and would fit the stepping-stone approach to applied research in the domain of computer technology; namely a series of research projects, each requiring increasing levels of technical and managerial know-how, and oriented toward a goal of national interest.

3. THE POLICE MESSAGE SWITCHING SYSTEM

The Police Message Switching (PMS) System has been designed and implemented by ELBIT Computers Ltd., according to the specifications of the Communication Division of the Israel Police. The system operates on a minimal hardware configuration, consisting of two CDC 1700 mini-computers each having 16 K of sixteen bit words. The PMS uses CDC's Mass Storage Operating System (MSOS), although only essential portions of it were retained. This yielded a compact system, with 8 K of core resident software during execution.

The PMS is a fully automatic, duplexed Store and Forward message switching system, capable of simultaneously serving up to sixty-four geographically dispersed subscriber stations (see Appendix II) with plans to expand service to eighty subscribers. Typically, stations operate in full duplex mode, using two teleprinters: one for message transmission and the other for message capture. If one of the teleprinters should become inoperative, the other one can both send and receive telegrams. The majority of the teleprinters in the existing network operate at fifty Baud transmission rate, with a limited number operating at one hundred Baud.

Telegrams are composed of four basic parts, as follows:

Leader
Header (message number, originating station name, number of destinations destination list and the number of copies for each teleprinter, routing priority)
Text (the message to be delivered)
Trailer

Messages are accepted and delivered in telegram format (see Appendix III) and their flow may be controlled by a priority mechanism. Both CDC computers receive incoming telegrams and record them in their respective queues. One computer, the active one, delivers telegrams and advises the other processor over a low-speed line each time a telegram is sent. This information enables the passive processor to mark its queues appropriately. When the active computer becomes disfunctional, the passive processor in effect takes over the teleprinter network, continues operating in telegram capture mode, and commenc transmission of queued and undelivered telegrams.

The major functions of the PMS may be summarized as follows:

1. telegram capture and queuing according to three priority classes,
2. provision of up to ten mailing lists in order to facilitate limited c general broadcasts,
3. copies of previously delivered messages may be retrieved from archival (history) storage, which has 1.5 million character capacity,
4. in case a destination can not be reached due to line or teleprinter malfunction, all associated telegram traffic can be routed to another nearby subscriber,
5. miscellaneous Network Controller functions, such as periodic status reports on queues and transmission lines.

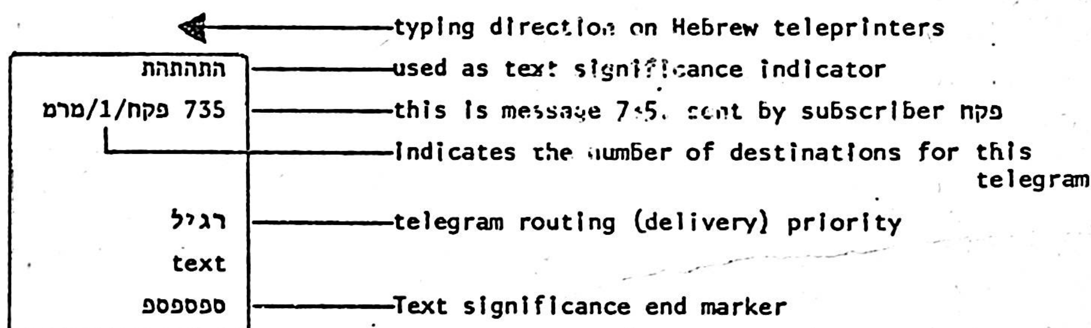
4. THE DPC SOFTWARE: OVERVIEW

The present on-line teleprocessing requirements of the Israel Police Data processing Center are mainly for inquiries and some file updates, as opposed to Remote Job Entry and interactive program development. In order to save time and manpower in implementing a multi-tasking, multi-application on-line system, DP management at the Israel Police selected IBM's CICS/VS program product as the teleprocessing monitor for the on-line system. CICS is an application program from the point of view of the operating system, and it coordinates all on-line activities within the DPC. A rich set of transactions have already been developed under CICS to assist various branches of the Israel police in their daily tasks. All technical work relating to application design and implementation has been performed by the staff of the Israel Police, and was carried out without any direct connection to the Network Project.

5. A HISTORICAL PERSPECTIVE ON THE PROJECT

When the connection between the PMS and the DPC "sub-networks" was initially considered, the PMS was already in advanced stages of testing. The DP "network" was at the time limited to one local terminal submitting a single type of an inquiry to a solo-tasking, home-grown telecommunication program, and there were plans to design and to implement an extensive teleprocessing-based application system. The fact that at the time that the Police Network specifications were written the future DPC application was only loosely formulat■ provided a convenient measure of freedom in the Network design.

In general terms, the specification called for modifications to the PMS to enable it to communicate with a synchronous terminal (the IBM System/370 via the 3704 communication controller) by means of the IBM Binary Synchronous Communication protocol (Reference 1). The general philosophy of the PMS was left intact and the DPC was considered to be yet another subscriber, named מרם, to be communicated with via telegrams. For example message number 735 from subscriber named פקח (the PMS Network Manager) to the DPC may be sent as follows:



See Appendix III for a sample telegram.

In man-to-man telegram traffic <u>text</u> is in free format and is unrestricted in content. In case the telegram is destined for the DPC, <u>text</u> is interpreted as a transaction or inquiry to which the IBM computer is expected to respond. The transaction must have the same form as it is entered by the DPC CRT terminal operators.

Telegrams received from the DPC cannot be directly routed to the originating teleprinter, since immediately after it sends an inquiry to the PMS it is "de-coupled" from the inquiry and is free to send and to receive other

telegrams. This consideration had a significant implication on the Network design, since it ruled out conversational inquiries, which would have required that teleprinters be locked onto transactions until "session completion' The framework of the PMS philosophy thus demanded that inquiry responses by queued in the originator's "Send" queue according to the original telegram's routing priority.

6. THE PHYSICAL CONNECTION BETWEEN THE TWO COMPUTERS

Two computers can exchange information in many different ways, including the following:

channel-to-channel (allows high speed information exchange between processors),
shared peripherals (e.g., discs or memory blocks, both of which allow communication by a "mailbox" type scheme, where messages are exchanged via "mail" delivered to pre-arranged rendezvous points),
tele-processing (connection is via standard teleprocessing facilities, namely modems and communication lines).

Due to the dissimilarity of the two computers, the shared peripherals approach was not pursued. A design based on a channel-to-channel interface was investigated; however, it was rejected for the following reasons:

- the estimated traffic load did not require a high bandwidth communication link,
- a hardware interface was not readily available to connect the CDC 1700 to the IBM System/370 Standard Channel Interface and the supporting software was assumed to be non-trivial,
- it was felt that in a mixed vendor environment, the clean system-to-system interface provided by modems is an invaluable asset during trouble shooting.

Teleprocessing has its own innumerable technological variants from which one must pick an appropriate solution to one's application. Since the PMS already supported terminals with teleprinter-type characteristics, some experts suggested that the two systems be directly coupled, with each thinking its partner was a teleprinter. This would have allowed a modem-less interconnection since the two systems would be even electrically compatible. A brief exercise in arithmetic showed that the simplicity of this approach would be paid for by the extremely poor performance of the resulting system. The solution eventually adopted was to connect the PMS and the DPC systems via 2400 Baud modems, with the PMS simulating the behavior of an IBM System/370 model 30 computer under the Binary Synchronous Communication Protocol (BI-SYNCH). The present link between the two computers is via an in-house communication line 100 meters in length (see Appendix I). Since there is only one

communication node at either end of the line, a point-to-point (contention) discipline was adopted, with the CDC processor considered to be the master. Performance considerations on the CDC side led to adoption of the ASCII, as opposed to the EBCDIC transmission code.

At the time when the project was initiated there was no controller available on the CDC 1700 to support directly BI-SYNCH, and consequently this necessitated addition of a suitable "driver" to the original PMS. In order to simplify the software effort, only a subset of BI-SYNCH was implemented. The resulting software can transmit only a single block (the telegram) to the IBM System/370, and can receive any number of blocks, since the length of responses cannot be determined in advance.

7. THE P^3 PROTOCOL

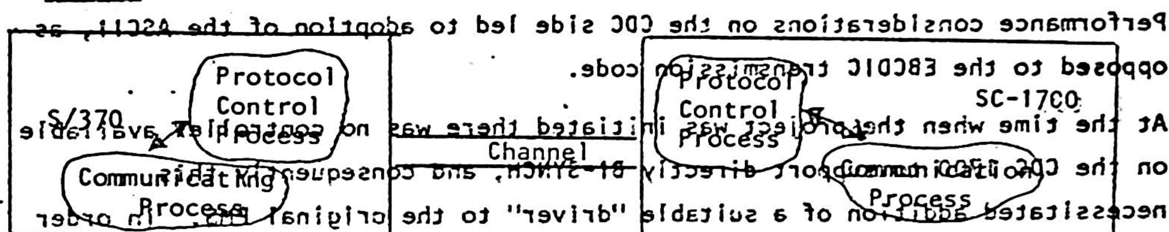
When two or more processes or processors desire to communicate with each other, they must conform in their interaction to a minimal etiquette, that is to a set of pre-arranged communication or "social" conventions. These generally cover the timing, format, sequence and encoding of messages sent by the processes across some information conductor, called a communication link, or channel. The "social contract" which cooperating communication processes adhere to is called a protocol in the vocabulary of computer communications.

At the time the project was designed the IBM product line supported the Binary Synchronous Communication (BI-SYNCH) protocol. Since it did not appear to provide sufficiently tight control over the processor-to-processor traffic, a simple Processor-to-Processor Protocol (P^3) was designed by ILTAM. Since the P^3 design was designed to meet the requirements of a specific system, it is not sufficiently general to solve the numerous problems encountered in the process-to-process communication on generalized, homogeneous computer networks.

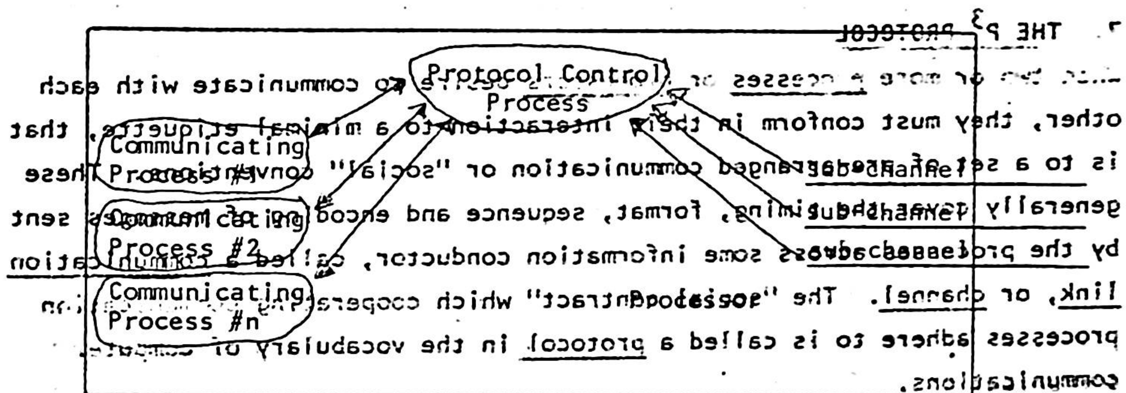
Information passed from processor to processor must adhere to the following general format:

Protocol Prologue	Text	Protocol Epilogue

A protocol control process is associated with each processor as follows:

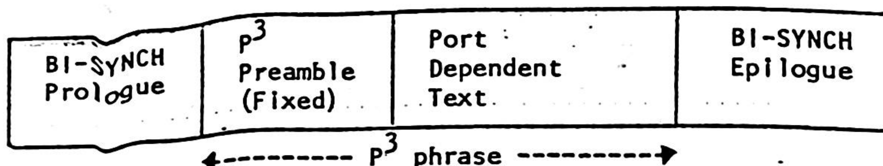


Conceptually, the processor-to-processor information duct is divided into a number of function-specific sub-channels called ports which is used by the processes of either half of the system as follows:



The Processor-to-Processor Protocol (P^3) can be thought of as composed of two parts, namely the carrier (BI-SYNCH) and the modulation (P^3). The former, like a carrier wave in radio communication, is the vehicle which enables messages to be transmitted from one processor to the other and the latter imparts the information value to the carrier. Together the P^3 protocol enables two processors to communicate in a reliable manner. The carrier enables the processors to talk to each other and the modulation enables them to understand each other. The carrier adheres to the Binary Synchronous Communications procedures as supported by standard IBM hardware and software. The 1700's protocol controller must, therefore, simulate these functions by software.

One transmission unit between the two systems is called a block. There are two kinds of blocks, namely line control and data blocks. The former are strictly BI-SYNCH control sequences, such as line bidding sequences, acknowledgments, synchronization etc., and are composed entirely of BI-SYNCH control characters. The latter have the following general format:



The structure of P^3 phrases is outlined in the present section. BI-SYNCH is described in detail in the appropriate IBM reference publication.

The format of the P^3 Preamble is as follows:

'TL17'	PORT NUM.	MSSG NUM.	BLOCK NUM.	PRTY	LENGTH	LOCK

PORT NUM. is a logical abstraction and in essence it uniquely identifies ports, or subchannels created by a "multiplexing" scheme on the inter-system link. The port concept was introduced since at the time of the preliminary system design, a general system-to system communication framework was envisioned, including file transfer between the PMS and DPC. The abstraction of ports simplified visualization of interaction between the systems. Presently the following ports are implemented:

1 - CDC to IBM traffic
2 - IBM to CDC traffic
4 - CDC to IBM re-connect
5 - unsolicited IBM to CDC traffic (that is not an inquiry response)

The functions of fields MSSG NUM, BLOCK NUM, PRTY and LENGTH are straight-forward, and are the message number of a telegram (internally assigned), the number of a block within a message, telegram priority in the PMS and the P^3 phrase length. LOCK is an interlock mechanism and indicates the following message states:

0 - additional blocks for the telegram follow
1 - last block of a telegram, additional telegrams follow (some inquiries are sliced into telegrams of smaller size, to prevent excessive monopolization of teletypes. In rare cases inquiry responses take up to ninety minutes to print out on slow teleprinters)
2 - last block of last telegram in response to an inquiry. The PMS retains the inquiry until this indicator is received, thus in case the two systems are disconnected before a response is fully delivered, the PMS will re-transmit the inquiry after it manages to re-connect. This indicator also re-opens port 1, and allows passage of another inquiry toward the DPC.

8. THE VIRTUAL TERMINAL FACILITY

Good engineering is characterized by clean, uncomplicated interfaces, with all complexity hidden from outside view. In the Israel Police Network it was realized that there was the ever-present danger of introducing irreversible future restrictions through dispersal of terminal dependent sensitivity amidst the growing population of application modules. Consequently, an early decision was made to localize all terminal dependencies, which could then be hidden from the view of application programs. The objective could have been met by a series of modifications to key CICS modules; however, this was not considered to be practical, mostly due to a well-founded fear of release dependence and uneasiness about altering proprietary software.

Despite the care taken in the design of the PMS system external interface to CICS, the CDC 1700 facing I/O port in the DP computer can hardly be treated by CICS application programs in the same manner as they look upon DPC CRT-s. The reasons for this are numerous, among which the following are typical:

- non-standard P^3 prefix,
- teleprinter line size (63 characters as opposed to 80 on 3270-s),
- control character differences between teleprinters and 3270-s,
- lack of a rich character set on the teleprinters,
- size restrictions on P^3 blocks (240 text characters) and telegram lengths (approximately 1800 characters),
- lockout of certain "long", low-priority telegrams during prime shift.

In view of the preceding, the earliest design specification already formulated a ground rule, explicitly stating that all system components without the appropriate "need to know" should visualize virtual terminals, thus permitting terminal or device independent application code generation. The abstraction was to be realized through the services of two programs which share among them the burden of device simulation and cater to non-standard PMS features.

In order to provide the necessary degree of isolation of application programs from terminal characteristics, the connection between these two resources was explicitly "de-coupled" by the two cooperating interface modules. This act served to create the illusion of a "virtual terminal": an abstract resource from which application programs obtain messages and to which they can route information. This facility is the only means available to the application modules in the DPC who wish to avail themselves of terminal input/output service

Various types of virtual terminals may be defined by means of appropriate macros in the Virtual Terminal Simulator. Virtual terminals may be assigned features found in the real terminals available within the network, or may be endowed with non-standard terminal facilities. Virtual terminal input/output requests may be delivered to a real terminal, and alternately text may be filed in scratch storage for later retrieval, or "screen-flipping" by terminal operators. It is also possible to gate output requests to (and receive input from) a lower level "non-standard" virtual terminal simulator program.

The following is a brief list of typical functions performed by the Virtual Terminal Facility:

- The active CDC 1700 processor communicates with CICS via a transaction. The leftmost sixteen symbols of the transaction carry control information (the P^3) and the remainder is a telegram addressed to the IBM computer as entered by the teleprinter operator. The telegram is composed of control information and text, and the control symbols in fact envelope the true transaction which is to be processed by the appropriate application module. Thus the original CICS transaction is a carrier for the real transaction to be processed. There is therefore the task of "opening the envelope", locating the intended transaction and the associated application module, and presenting the transaction to the module.
- Application modules convey their terminal input/output requests to one of the two interface modules responsible for Virtual Terminal Support. That module retains line-by-line output requests, until a "page" of data has been formed. In case the text is directed to the PMS, page formatted data is presented to the other interface module, which converts arbitrary length text into telegrams acceptable to the PMS. The telegrams are further processed, to affix the P^3 prefix to the units of information exchanged between the two computers.
- Telegram formation is restricted by a pragmatic rule limiting text length to approximately 1800 characters, in order to prevent monopolizing the essentially low-speed destination teleprinters. Long text fragments are "sliced" into telegrams of manageable length, with an attempt being made to break text at logical boundaries.

- Guarding against telegram loss in a system of this complexity is by itself a design problem. Rather than to surround the already far from simple CICS with an elaborate checkpoint scheme, the P^3 specifies a "ping-pong" type interaction between the two processors, in which the CDC 1700 presents a telegram to the IBM 370, holds the telegram in its 370 bound output queue, receives all related responses and releases the telegram from its queue only once the IBM computer advises it that telegram processing has been completed. This in fact amounts to a process-to-process level end-to-end acknowledgment.
- The PMS performance is in some sense a function of the load its subscriber population can exert in a given unit of time. Once a computer, such as the IBM System/370 model 145 becomes a "subscriber", it can in theory flood the system with telegrams and may overload portions of the PMS network, unless some form of "flow control" procedure is adhered to. A simple control mechanism has been designed, which bars "long telegram" processing during peak hours, except in very high-priority cases, when the flow-control defense mechanism may be bypassed. Flow-control is one of the tasks of the interface modules.
- Non-trivial system-oriented errors are intercepted, and appropriate messages are sent to the PMS Network controller's teleprinter, in telegram format.
- Network Controller notification whenever the computers establish initial communication.
- Each PMS teleprinter is treated as a tributary station to the single 2400 Baud BI-SYNCH line connecting the computers. In particular, integral CICS performance measurement and accounting routines are provided with the capability to reflect work-load down to the tributary station level.

The essentially limited teleprinter keyboard facilities are functionally extended in two orthogonal ways, namely:

- "control keys" are simulated for text delete, line delete, backspace functions
- the severely limited teleprinter character set is extended by providing a means of entry for unrepresented symbols, such as the Hebrew final (sofit) characters, or letters of the Latin alphabet.

The extensions are realized by appropriate key-strokes on the teleprinters, to be interpreted by the Virtual Terminal Facility.

The following control sequences are typical:

?0 · Line Delete	?מ ם (Mem Sofit)
?5 Backspace five	?+ Latin Case Entry
?9 Text Delete	?? ?

The Virtual Terminal Facility proved to be a natural place to provide the various housekeeping or editing functions inevitably found in on-line systems, such as

- trailing blank stripping on output,
- automatic margination/tabulation of free form text with a choice between left (for English) and right (for Hebrew) text alignment.

9. ERROR RECOVERY CONSIDERATIONS IN THE DATA PROCESSING COMPUTER

CICS provides several levels of error recovery, namely:

General level: The action taken is mostly logging and signalling terminal operators of error occurance. More specific error recovery can be performed by customized user exits.

Specific Level: Each application program may establish an error recovery environment, which allows it to perform error recovery.

In case of errors related to the PMS, some are resolved by the P^3 protocol, which guards against inconsistent transmission sequences and prevents telegram loss. In the section describing the P^3 protocol, it was described how the PMS retains the last telegram transmitted to the DPC until it receives the "End-to-End" acknowledgment, which signals that the processing has been completed and that the original telegram may be discarded.

Two problems still had to be solved in order to create a smoothly-operating system. When CICS detects certain types of terminal I/O errors, it routes an appropriate message to the terminal associated with the process during whose execution the error was detected. The message, of course, does not conform to the P^3 protocol, since the CICS system modules do not make use of the Virtual Terminal Facility. In order to prevent transmission of garbled messages to the PMS, the PMS Interface Module (one of the two Virtual Terminal Facility modules) establishes an error recovery environment, intercepts error conditions and routes error indicators to the PMS Network Controller in the form of telegrams.

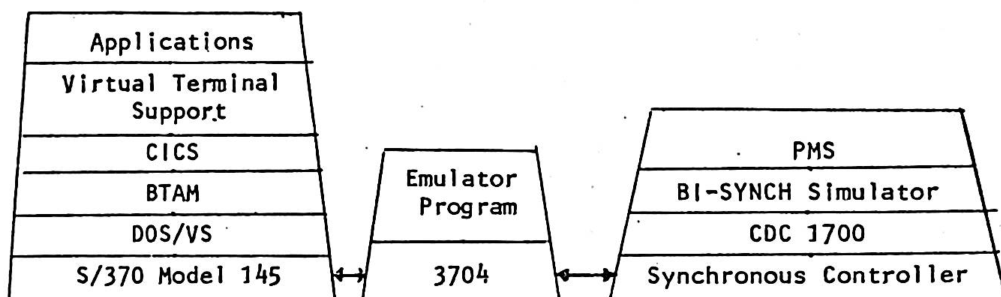
Another error recovery problem arises when a physical malfunction occurs in the communication link between the two computers. In such circumstances a "deadlock" situation may develop, with the PMS attempting to re-establish a connection, while CICS continues with the interrupted discourse. To avoid this possibility, the CICS Terminal Error Program was altered, in order to "kill" PMS-associated processes in case of certain error situations. Dictatorial though this policy is, nothing is lost, since the P^3 protocol guarantees that the last transaction is re-transmitted after the processors establish contact again.

10. INTEGRATION TESTING

Any project in which more than one organization participates must allocate non-trivial resources for system-level integration testing. This is perhaps the most valuable lesson which may be learned from the Israel Police Network project. Integration testing was preceded by simulated sub-system tests conducted by ILTAM and ELBIT independently in Jerusalem and in Haifa, respectively. ILTAM's tests were based on punched card based simulated inter-system traffic. This was performed under CICS control and tested all aspects of the DPC sub-system, with the exception of real (as opposed to simulated) terminal communication. In parallel, two CR17 (CDC 1700 compatible) CPU-s were set up in the ELBIT Haifa development laboratory, with the computers simulating the PMS and DPC and communicating in BI-SYNCH.

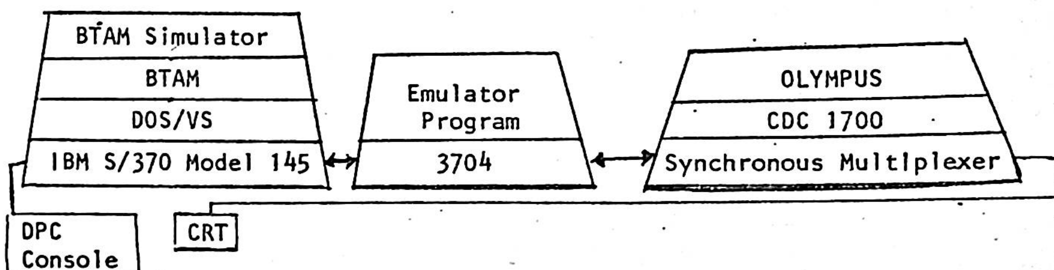
The success of pre-integration testing depends on the ability to construct an accurate analogue of the real world situation to be mimicked. Thus, despite the fact that both the DPC and PMS sub-systems appeared to function under simulated conditions, they failed to communicate with each other after hook-up. The basic debugging tool turned out to be the IBM 3704 Line Trace utility, a program capable of recording and displaying all inboard and outboard traffic at both the line and channel sides of the controller. The trace output was rather difficult to read, but demonstrated that information was traversing the system for a brief period of time after connection took place between the two systems.

The system was relatively difficult to debug due to the large number of hardware and software layers present as well as because of timing problems. An approximate diagram of the layering is as follows:



It was decided to simplify this as much as possible. ILTAM wrote a small BTAM program, which enabled fine control over BTAM via the DPC console terminal. It provided the ability to issue BTAM commands such as Read Initial, Read Continue, Write Initial, Write Continue and EOT.

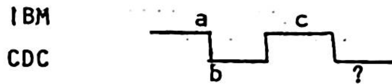
ELBIT also replaced the PMS with OLYMPUS, a simple-to-use system designed for hardware fault diagnosis. The OLYMPUS CRT was placed next to the DPC console and OLYMPUS could thereby be instructed to send BI-SYNCH command sequences such as ENQ, ACK-0, ACK-1, EOT, NACK, RVI, WACK or text. This testing configuration permitted dynamic display on the CRT of all significant text traversing the communication line. The resulting system had the following simplified appearance:



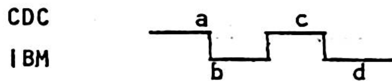
Debugging consisted of repeated attempts to send various BI-SYNCH phrases (e.g. ENQ/ACK-0/text/ACK-1/.../EOT), noting the information on the OLYMPUS CRT and checking the content of the DPC buffer after DPC read operations. One of the main problems initially faced turned out to be that a faulty parameter setting in the 3704 EP permitted parity-bit transfer into S/370 memory as the zero-th data bit in each byte.

During the course of OLYMPUS-based integration testing, ILTAM formulated its own version of Turing's "Imitation Game". The objective was to try to learn from the IBM 3704, its EP and BTAM the legal responses to various BI-SYNCH

sequences to which the response was unclear. If the interchange in doubt was



then in the framework of the Imitation Game the roles were reversed as follows:



with d forming the learned legal response to the a/b/c chain.

The communication problems were more or less resolved before skill was developed at the Imitation Game. The DPC was again loaded with CICS and testing progressed with "manual BI-SYNCH" on the CDC 1700. Some problems steadily persisted, in particular with the EOT phrase. As these problems were resolved OLYMPUS was replaced by the full PMS and gradually the system became operational.

11. CONCLUSIONS

The system recently began operation in production mode, and is providing reasonable response time. As experience will be developed in the use and operation of the Network, three possible changes will be investigated, as follows:

-- modification to the CICS PMS Interface Module (one of the virtual terminal simulator routines) and of the appropriate PMS programs in order to permit multi-tasked and interleaved processing and transmission of transactions originating with the teleprinter network. The objective of this change would be to improve teleprinter response time.

-- provision of enhanced teleprinter terminal and keyboard facilities such as handling "bright" fields, which are used to highlight information on the IBM 3270 CRT-s and which are left "dim" when routed to the teleprinters.

-- feasibility of conversational transaction implementation for the teleprinters which are presently de-coupled from telegrams immediately after their receipt by the PMS.

Perhaps the most important conclusion is that despite the present de facto policy of de-emphasizing software development in Israel, complex software can and should be developed here in the future.

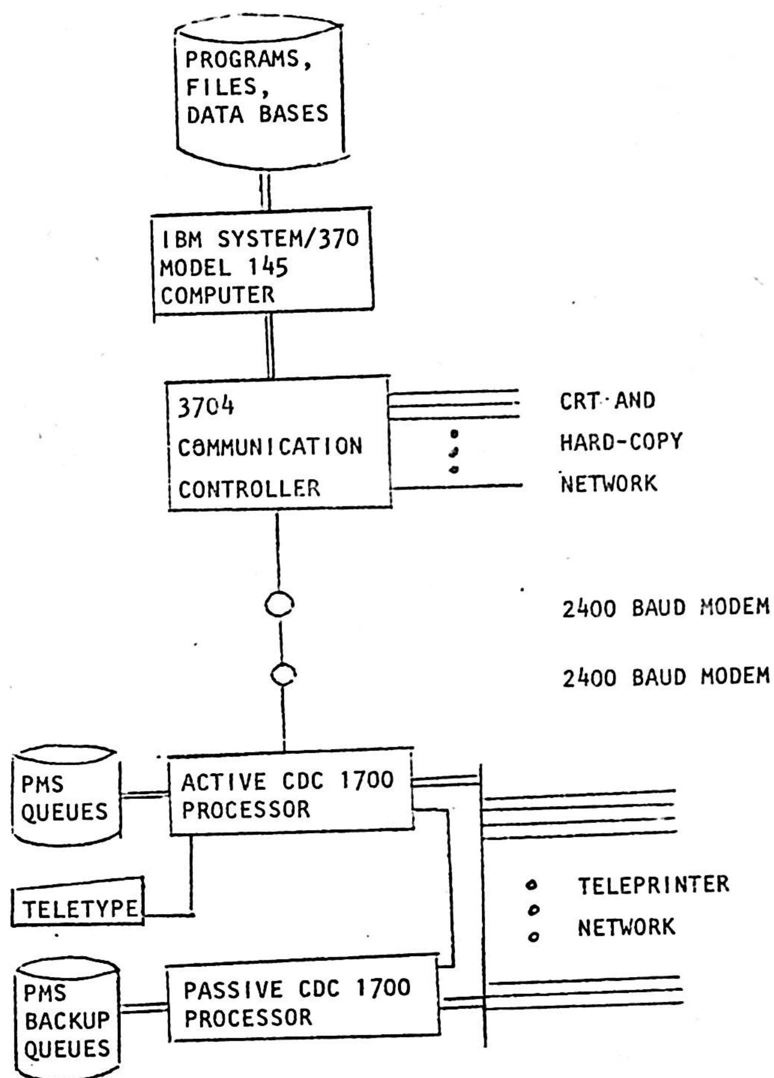
References

[1] General Information - Binary Synchronous Communications
IBM Form Number GA27-3004

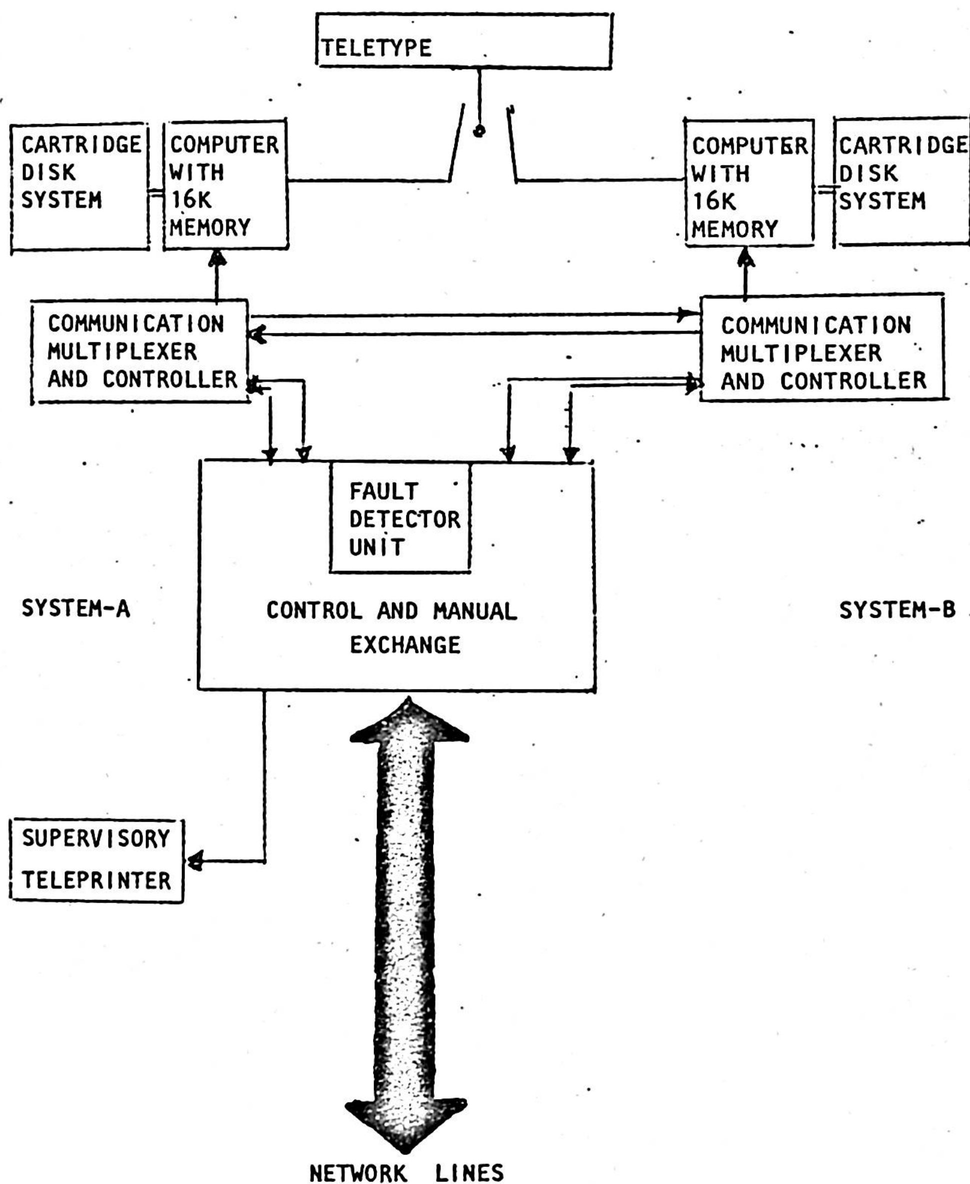
Acknowledgements

We would like to express out thanks to Mr. Ronnen and Mr. Sharon of the Israel Police Data Processing and Communication Divisions for their unceasing support. This project could not have been completed without Mr. R. Guzman and Mr. M. Kikayon, who conceived the Network along with the Israel Police Headquarters staff and whose encouragement was never lacking. Thanks are due to Mr. Itzhak Raz (ELBIT), who spent many nights helping us debug hardware problems.

APPENDIX I: NETWORK SCHEMATIC DIAGRAM



APPENDIX II: THE POLICE MESSAGE SWITCHING SYSTEM



APPENDIX III: SAMPLE IBM SYSTEM/370 BOUND TELEGRAM AND RESPONSE

(A)

התהתהת
17 פקת/1/מרמ
רגיל
רסבג89
כבסבכב

(B)

פקח 080
010 ברמ/1/פקח
הוב
+001/017+

```
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 אבגדהוFטי 0123456789 אבגדהוFטי 0123456789 אבגדהוFטי
0123456789 א
```

2140.10

כבסבכב

09/76 012141 012140

(A) TELEGRAM NUMBER 17 SENT TO THE IBM SYSTEM/370. THE TRANSACTION REQUESTS SENDING OF A TEST PATTERN

(B) PART 1 (AND LAST) OF THE SYSTEM/370'S RESPONSE TO TELEGRAM NUMBER 17.

על מהימנותם של קוי תקשורת נתונים בארץ

מ. להב - אילתם - האיגוד לתכנון ולמחקר בע"מ

I מבוא

אין שום ספק שתקשורת נתונים משחקת תפקיד חשוב וגובר בכל הקשור בתכנון והפעלת מערכות מידע ממוכנות בארץ. אחד האספקטים החשובים של תקשורת נתונים, הוא מהימנותה של מערכת התקשורת, או ליתר דיוק תת-מערכת התקשורת שהיא למעשה רק מרכיב אחד של המערכת הכוללנית. כמעט בכל מפגש ציבורי של אנשי מחשבים בארץ, מתנהלת שיחה בנושא התקשורת, ולעתים קרובות מאד, שיחה זו מקבלת צורה של ביקורת לגבי שרותי התקשורת הנתנים בארץ. בין היתר טוענים שהקווים אינם מהימנים, וזה גורם לאי-רצון מסויים של מוסדות להתבסס על תקשורת נתונים מהחשש שאי-מהימנותם של הקוים תפריע להפעלה סדירה של המערכת המתוכננת. לאור חשיבות הנושא, ערכה אילתם סקר קצר בנדון, ותוצאותיו מובאות במסמך זה.

לא היה באפשרותנו לערוך מפקד אשר יכלול את כל מאות הקווים הפועלים בארץ, או אפילו את רובם; אלא בחרנו מדגם של אוכלוסית הקווים המעניין אותנו במיוחד, קוי תקשורת המופעלים ע"י משרדי הממשלה. מדגם זה, כולל 30 קוים וציוד הקצה המחובר אליהם, ותקופת הזמן עבורו יש לנו נתונים הוא חמישה חודשים לגבי רוב הקוים (22), וחודש וחצי לגבי היתר (8). בסך הכל גודל המדגם הוא 117 "חודשי-קו". אנו סבורים שמדגם זה מייצג את המצב הכללי, מכיוון שקוים אלה נמצאים על הצירים המרכזיים של העברת נתונים, לפחות בסקטור הממשלתי, והם : ירושלים - תל-אביב, ירושלים - חיפה, ירושלים - אשדוד, ובתוך ירושלים. כלול בתקופת המדגם (ספטמבר 75 - פברואר 76) חודשי החורף שהם אמורים להיות החודשים "הקשים" ביותר מבחינת תקלות בקוים.

מקורות המידע שלנו לגבי סקר זה, היו יומני התקלות של המרכז למיכון משרדי (מל"מ), ושרות עיבודים ממוכנים (שע"מ). אנו אסירי תודה לאנשי מתקנים אלה על שיתוף הפעולה באיסוף הנתונים שבלעדיהם לא יכולנו לערוך סקר זה.

<u>אזהרה</u>: כפי שניזכר לעיל, מקורות הנתונים של סקר זה היו יומני התקלות של שני מיתקני מחשב ממשלתיים. ייתכן שלא כל התקלות שערו במשך פרק הזמן הנסקר נרשמו, וזאת עקב הדיווח והרישום הידניים. כמו-כן ייתכן שערו תקלות נוספות בקוים בשעות הלילה או בשבתות בהן לא הופעלו המסופים, ולכן אין ידיעות על תקלות מסוג זה.

הדרך האידיאלית לערוך סקר כזה הוא, כמובן, לחבר מכשיר המשגיח באופן תמידי על ביצועי הקוים. אך ללא מכשיר כזה חייבים להשתמש ביומני התקלות במקור הטוב ביותר לנתונים. יתר על כן מדגם גדול יותר (או ע"י הגדלת מספר המיתקנים והקוים שמופיעים במדגם, ו/או ע"י הארכת פרק הזמן הנסקר) יתן תמונה מדויקת יותר מאשר סקר זה.

לפי ממצאי הסקר, מערכות תקשורת הנתונים בארץ, זוכות לציוני אמינות גבוהים. שלושים המערכות שבדקנו (כאשר מערכת פרושה קו נל"ן וציוד הקצה המחובר אליו - מודמים, מסופים ויחידות בקרה), פעלו באורח תקין % 97.4 מהזמן. רמת מהימנות זו משתווה בחיוב רב עם הסטנדרטים המקובלים לגבי המחשבים עצמם (בדרך כלל % 90 או % 95). קשה לקבוע במדוייק אלו מרכיבים היו אמינים פחות ואלו אמינים יותר, משום שבדרך כלל יומני התקלות לא כללו מידע זה, אך אפשר לציין שכ-% 50 מהתקלות ארעו בקוים עצמם, כ-% 36 בציוד הקצה, וביתר % 14 מן המקרים לא עלה באפשרותנו להצביע על המקור המדויק של התקלה. לפי נתונים אלה, אפשר לחשב את הזמן הממוצע בין תקלות (MTBF) למרכיבים העיקריים כדלקמן :

טבלה 1 - מהימנותם של קוי תקשורת נתונים

מרכיב	מספר תקלות לקו לחודש	זמן ממוצע בין תקלות (שעות)
מערכת (קוים + ציוד קצה)	1.15	630
קוים בלבד	0.58	1261
ציוד קצה	0.41	1760
תקלות בלתי ידועות/ אחרות	0.16	4969

טבלאות 2, 3, ו-4, מסכמות את התפלגות התקלות על פני הקוים המחוברים למל"מ. מטבלה 2, אנו רואים כי אף קו אחד לא יצא נקי ממכת התקלות, אך שיעור התקלות איננו אחיד לגבי כל אוכלוסית הקוים. שיעור התקלות הממוצע במשך חמישה חודשי המדגם, היה 5 תקלות לקו. מטבלה 2 אנו רואים כי % 73 מהקוים סבלו תקלות בשיעור ממוצע או פחות, כאשר רק 6 מתוך 22 הקוים (% 27), סבלו תקלות בשיעור מעל לממוצע. מבט בטבלאות 3 ו-4 מראה כי תקלות בקוים עצמם מפרידות בין קוים טובים לקוים גרועים יותר מאשר ציוד הקצה. הקוים הם יותר משתנים מאשר ציוד הקצה. במילים אחרות, הסיכוי לקבל קו גרוע הוא יותר גדול מאשר הסיכוי לקבל פריט של ציוד קצה גרוע.

מצאנו רק מאפיין אחד של הקוים הגרועים, והוא שקוים שמגיעים לחיפה, סובלים שיעור תקלות כפול מקוים אחרים. לא מצאנו כל קשר בין מאפיינים אחרים, כגון מהירות, קו רב-קצוותי (multi-drop) לעומת קו נל"ן רגיל, אזורים אחרים בארץ (בפרט אזורים בתוך תל-אביב) וכו'.

טבלה 2 - התפלגות תקלות בקוי תקשורת מחוברים למל"מ

מספר תקלות	0	1-3	4-6	7-9	10-12	13-15
מספר קוים	0	9	7	2	2	2
אחוז הקוים	0	41	32	9	9	9
אחוז מצטבר של הקוים	0	41	73	82	91	100

טבלה 3 – התפלגות תקלות קו בקוי תקשורת מחוברים למל"מ

מספר תקלות	0	1–3	4–6	7–9	10–12	13–15
מספר קויה	7	9	4	1	0	1
אחוז הקוים	32	41	18	4.5	—	4.5
אחוז מצטבר של הקוים	32	73	91	95.5	95.5	100

טבלה 4 – התפלגות תקלות בציוד קצה בקוי תקשורת מחוברים למל"מ

מספר תקלות	0	1–3	6–4	7–9	10–12	13–15
מספר קוים	7	12	2	1	0	0
אחוז הקוים	32	55	9	4	—	—
אחוז מצטבר של הקוים	32	87	96	100	100	100

III זמן חיסול תקלות

לעומת האמור לעיל, אי-אפשר לפרש רמת אמינות גבוהה כסימן של מצב משביע רצון לחלוטין. על המוסד המבקש להפעיל מערכת תקשורת נתונים בארץ להתכונן לכ-19 שעות תקלה על כל קו שהוא מפעיל. זמן תקלה זה נובע מ-1.15 תקלות בקו בממוצע בחודש וזמן תיקון ממוצע (MTTR) של 16.5 שעות. קשה להתייחס לממוצעים, מכיוון שהסטיה של זמן חיסול תקלות הוא גדול (סטית התקן היא 24.4 שעות). אפשר לסכם את משך התקלות בטבלאות 5 ו-6

טבלה 5 - התפלגות זמן חיסול תקלה

זמן חיסול תקלה	אחוז התקלות שחוסלו תוך פרק הזמן	אחוזים מצטברים
עד שעתיים	0.16	0.16
שעתיים עד שמונה שעות	0.41	0.57
שמונה שעות עד 24 שעות	0.43	0.84
מעבר ל-24 שעות	0.16	1.00

טבלה 5, המתבססת על תיקון תקלות מכל הסוגים ע"י כל גורמי השרות, מראה שיש סיכוי סביר (84 %) שהתקלה תחוסל תוך 24 שעות, אך קיים סיכוי די גדול (16 %) שמשך התקלה יעלה על יממה אחת. התקלה הארוכה ביותר שהופיעה במדגם, חוסלה כעבור שישה ימים.

טבלה 6 - הצטברות זמן תקלה לקו לחודש

זמן תקלה לקו לחודש	אחוז התקלות שהתמשכו במשך הזמן	אחוזים מצטברים
0 - 10	0.47	0.47
11 - 20	0.23	0.70
21 - 30	0.13	0.83
31 - 40	0	0.83
41 - 50	0.03	0.86
51 - 60	0	0.86
61 - 70	0.03	0.89
71 - 80	0	0.89
81 - 90	0	0.89
91 - 100	0	0.89
101 - 110	0.03	0.92
111 - 120	0	0.92
121 - 130	0	0.92
131 - 140	0.07	0.99
141 - 150	0	0.99 *

* לא היו מקרים מעל ל-140 שעות, אלא שהמספרים לא מסתכמים ל-1.00 בגלל בעיות עיגול.

טבלה 6 מסכמת את התפלגות זמן התקלה לקו לחודש. יש לזכור שתקלות בקו בדרך כלל משתקות את כל הקשר על הקו, כאשר תקלות בציוד קצה יכולות לשתק רק מסוף אחד מבין מספר מסופים מחוברים לקו. המידע שעמד לרשותנו במדגם לא אפשר לנו להפריד בין שיתוק חלקי ושיתוק מוחלט. בנוסף יש להוסיף את זמן התקלה הנובע מתקלות במערכת המחשב המרכזי, כגון נפילות מערכת ההפעלה, תקלות במחשב עצמו, תקלות בתכנת היישום, תקלות בחשמל וכו'.

מבחינת רמת השרות המתקבלת מאת גורמי השרות הנוגעים בדבר, מצאנו זמני תקלות ממוצע כדלקמן :

טבלה 7 - זמן חיסול תקלה ממוצע לפי סוג התקלה

ה מ ר כ י ב	זמן חיסול תקלה ממוצע (שעות)
מערכת (קוים + ציוד קצה)	16.5
קוים בלבד	18.5
ציוד קצה	13.8
בלתי ידועים / אחרים	15.5

מנתונים אלה, אפשר להסיק את המסקנה שמשרד התקשורת מגיב בצורה פחות טובה מאשר ספקי הציוד הקצה. ההפרש בין זמן חיסול התקלה הממוצע בין תקלה בקוים לבין תקלה בציוד הקצה מגיע לכ-5 שעות.

טבלה 8, מסכמת את רמת השרות המתקבלת מאת גורמי השרות השונים. מבחינת תגובה מידית (עד שעתיים) משרד התקשורת הגיב בצורה יותר טובה מאשר ספקי הציוד הקצה. אחוז גבוהה של תקלות בלתי ידועות/אחרות שתוקנו באופן מיידי, מוסבר בעיקר ע"י מספר גדול של תקלות מסוג זה שחלפו בעצמן, כלומר תופעה של תקלות חולפות בציוד שלא דרשו תיקון ע"י גורם כלשהו. מבחינת תגובה תוך אותו יום ותגובה תוך יממה אחת, ספקי הציוד הקצה מגיבים באופן טוב יותר ממשרד התקשורת.

אם נגדיר כתקן שרות רצוי ש-99% מהתקלות החוסלנה תוך 24 שעות (כלומר שכמעט אף מנוי לא ישבת יותר מיום אחד בגלל תקלות טכניות), הרי שגם משרד התקשורת וגם ספקי הציוד הקצה לא עומדים במבחן, אם כי המצב יותר טוב לגבי הספקים של ציוד הקצה מאשר אצל משרד התקשורת.

טבלה 8 – השוואת רמת שרות של הגורמים השונים

מס' סד'	הגורם	מספר תקלות	זמן חיסול ממוצע (שעות)	סטית התקן (שעות)	אחוז התקלות שחוסלו			
					תוך שעתיים	תוך 8 שעות	תוך 24 שעות	מעל ל-24 שעות
1.	סה"כ	135	16.5	24.4	16	41 (75)*	27 (84)	16 (100)
2.	קוים	68	18.8	29.1	19	38 (57)	21 (78)	22 (100)
3.	ציוד קצה	49	13.8	14.6	8	45 (53)	37 (90)	10 (100)
4.	בלתי ידועים/אחרים	18	15.5	26.5	28	44 (72)	11 (83)	16 (100)

* אחוזים מצטברים

TV ממצאים אחרים

בחור "קוריוז", חישבנו את שיעור התקלות בחודשי החורף לעומת שיעור התקלות בחודשים האחרים, על מנת לאמת את ההרגשה המקובלת שחודשי החורף הם החודשים ה"קשים" מבחינת תקלות בקוים. להפתעתנו מצאנו ששיעור התקלות בקוים בלבד בחודשים דצמבר-ינואר, היה 0.51 תקלות לקו לחודש לעומת שיעור התקלות בחודשים ספטמבר-נובמבר שהיה 0.58. אין אנו טוענים שבחורף המצב הוא יותר טוב מאשר בחודשים אחרים, אך לפחות בשנת 1975 לגבי 22 קוי תקשורת המחוברים למל"מ, כך היה המצב. יש לציין שחורף 1975, היה יבש במיוחד וללא ספק עובדה זו מסבירה את התופעה.

V מסקנות

אפשר להסיק מספר מסקנות מנתוני הסקר שנערך על ידנו ומהערות אנשי המתקנים שאיתם שוחחנו :

(א) רמת המהימנות של מערכת תקשורת הנתונים שנכללו במדגם, היתה גבוהה (97.4 %) ומשתווה בחיוב לרמת המהמנות המקובלת לגבי מחשבים (95 % - 90 %).

(ב) על המוסד המבקש להפעיל מערכת תקשורת נתונים בארץ, להתכונן לכ-1.15 תקלות וכ-19 שעות תקלה לחודש על כל קו שהוא מפעיל. אפשר לשייך כ-50 % מהתקלות וזמן התקלה לקוים עצמם, כאשר היתר שייכים לציוד הקצה (36 %) ולתקלות שמקורן לא צויין במדוייק בחומר שעמד לרשותנו בסקר (14 %).

(ג) הרוב המכריע של התקלות, יחוסל תוך 24 שעות (84 %), אך קיים סיכוי סביר שהתקלה תמשך מעבר לזה עד למקסימום של שישים ימים (בנתוני מדגמנו).

(ד) רמת השרות המתקבלת מאת משרד התקשורת נופלת מזו של ספקי ציוד הקצה. שתי עובדות מציגות נקודה זו והם :

- זמן חיסול התקלה הממוצעת :

  18.5 שעות עבור המשרד, לעומת

  13.8 שעות עבור הספקים

- אחוז התקלות שחוסלו תוך 24 שעות :

  78 עבור המשרד, לעומת

  90 עבור הספקים

(ה) יש מקום לשקול אמצעי גיבוי מתאימים למסופים קריטיים. למשל, נתוני הסקר מראים שקו גיבוי על הציר ירושלים - תל-אביב שמחירו ביום כ-2,600 ל"י לחודש, מוצדק אם מחיר הנזק של שעת הפסקת קשר מגיע ל-275 ל"י ויותר.

(ו) משרד התקשורת מסוגל לספק קוים מהימנים ששיעור התקלות שלהם נמוך יחסית. על המנוי לדרוש החלפת הקו במקרה שיסבול שיעור תקלות גבוהה מן הממוצע.

(ז) קוים על הציר ירושלים - חיפה, נמצאו פחות אמינים מקוים על צירים אחרים שבמדגם. שיעור תקלות שנבעו מהקוים עצמם היה כפול מן הממוצע על קוים אלה.

(ח) מלבד נקודה ז' לעיל, לא מצאנו קשר הדוק בין שיעור התקלות על הקוים לכל מאפיין שהוא, לרבות: מהירות, נל"ן רגיל לעומת קו רב-קצוותי וכו'.

(ט) על מפעיל מערכת התקשורת לדאוג לאמצעי איתור תקלות לפחות לרמה של בדיקות ה-loopback המקובלות על המודמים הטובים הנמצאים בשוק. בדיקות אלו מאפשרות בקלות לאתר את מקור התקלה - קו, מודם בקצה המחשב, מודם בקצה המסוף, מסוף, או יחידה בקרה. ללא רכת בדיקות זו מפעיל התקשורת יסבול זמן תקלה מיותר.

(י) ישנן תקלות רבות שמתחילות להראות סימנים לפני תקלה מוחלטת. לכן כדאי לתכנן מערכת התראות אשר תצביע על מצבים לא תקינים בביצועי הקו וציוד הקצה בטרם יופסק הקשר כליל. יתר על כן ישנם מקרים רבים של תקלות חוזרות לאחר מסירת קו או פריט של ציוד קצה חזרה למנוי/צרכן לאחר תיקון. מערכת התראות, יכולה לגלות ציוד שלא תוקן כראוי על מנת למנוע מקרים אלה.

(יא) נתונים שלא פורסמו בדו"ח מטעמים שונים, מצביעים על הבדלים משמעותיים ברמת השרות המתקבלת ע"י שני המתקנים מהם קבלנו נתונים. אפשר לשייך לפחות חלק מהפרש זה למערכת היחסים שפותחה במשך הזמן בין אנשי המתקן לבין נותני השרות. ההתרשמות היא שמערכת יחסים טובה כזאת, היא גורם חשוב בכל הנוגע לקבלת שרות טוב.

הצעה להקמת רשת מחשבים אב-טיפוסית בישראל

מ. להב - אילתם - האיגוד לתכנון ולמחקר בע"מ

1. מבוא

מוגשת בזאת הצעה מפורטת להקמת רשת מחשבים אב-טיפוסית בישראל. הרשת המוצעת מבוססת על מרכזי חישוב אוניברסיטאיים ויעודה העיקרי הוא, לצבור ידע ולמצוא פתרונות לבעיות של הפעלת רשת מחשבים בארץ והשימוש בה, ע"י ביצוע פרויקט נסיוני. הצעה זו כוללת בתוכה נתונים מפורטים על מסגרתו ומטרותיו של הפרויקט, לוח זמנים, מבנה ארגוני, דוגמה של קונפגורציה אפשרית, אומדני תקציב ועוד.

התחום של רשתות מחשבים מהווה אחד מהתחומים החשובים ביותר מבחינת התקדמות טכנולוגית במיחשוב במשך עשר השנים האחרונות. עם התחלה צנועה באמצע שנות השישים כמספר קטן של פרויקטים מחקריים, תחום הרשתות התקדם לנקודה בה הן רשתות מסחריות והן רשתות מחקריות פועלות כיום באופן שוטף גם בצפון אמריקה וגם באירופה. אנו מגדירים רשת מחשבים כאוסף של מחשבים הממוקמים במקומות שונים, המחוברים באמצעות תת-רשת תקשורת על מנת לאפשר לצרכני הרשת להתקשר עם כל מחשב באמצעות מסופים, ועל מנת לאפשר התקשרות בין המחשבים לבין עצמם.

המטרה העיקרית של רשת מחשבים היא לספק שרותי מחשב (חמרה ותכנה) זולים ואמינים לקהל רב ככל האפשר, באמצעות תקשורת סיפרתית ומסופים. רשתות מאפשרות למשתמש במחשב להנות מהיתרונות לגודל של מחשבים גדולים וענקיים, להקטין את מידת הכפילות בתכניות ובאנשי תמיכה, ולהנות מהיתרונות של מחשב "בבית", הכל במסגרת מערכת אינטגרטיבית אחת. בקיצורו של דבר, רשתות מחשבים פותרות את בעית הביזור מול הריכוז של המחישוב ע"י מתן יתרונותיהן של שתי השיטות למשתמש. מקובלת הדעה בחוגים מקצועיים שרשתות מחשבים יהיו מרכיב מרכזי של "הדור הבא" של המחשבים.

המטרה העיקרית של הפרויקט המוצע, איפוא, היא לקדם את הטכנולוגיה של רשתות מחשבים בארץ. מכיון שהדרכים הבסיסיות לחיבור בין מחשבים כבר נחקרו ופותחו באופן נרחב במשך עשר השנים האחרונות, פרויקט זה יתבסס על טכנולוגיה קיימת. המדובר הוא ביישום טכנולוגיה זו בישראל ע"י חיבורם של מספר מרכזי חישוב אוניברסיטאיים ביחד על מנת ליצור רשת מחשבים צנועה, אשר תשמש תשתית חיה לצבירת ידע ונסיון בדבר ההפעלה והשימוש ברשתות מחשבים בארץ. הסביבה המתאימה ביותר לפרויקט מהסוג הזה היא כמובן הסביבה האוניברסיטאית, הודות לאופיה האקדמית/מחקרית. באופן מסורתי הרשתות הראשונות נבנו במסגרת אוניברסיטאית, מה גם שבישראל סקטור האוניברסיטאי מרכז בתוכו אולי את הריכוז הגדול ביותר של כח חישוב במדינה. מצוי בסקטור זה גם מאגר של כח-אדם מקצועי. שני האלמנטים הללו דרושים מאוד ע"י רשת מחשבים.

זאת לא הפעם הראשונה שמציעים הקמת רשת מחשבים בארץ. בשנת 1973 הוצע פרויקט דומה ע"י ה"ה פרופ' אלן רייטר מהטכניון וד"ר יהודה קלע ממשרד התקשורת. בתגובה להצעת רייטר-קלע נמצאה אז התענינות בקרב האוניברסיטאות וחוגים ממשלתיים (אגף התקציבים והמועצה העליונה להכוונת ענ"א), אך הפרויקט לא יצא לפועל; וזאת בעיקר מחמת חוסר גורם מרכז ומתאם הפרויקט.

עתה, שלוש שנים לאחר מכן, ההענות עוד קיימת בקרב האוניברסיטאות ובחוגים ממשלתיים, אפשר להצביע על ארבע עובדות לפתוח המעידות על כך, והן :

* הסכמה עקרונית בין כל המנהלים של מרכזי החישוב האוניברסיטאיים על חשיבות הפרויקט והבטחות של שיתוף פעולה בתנאי שהפרויקט יזכה למימון ממקורות מחוץ לתקציביהם.

* עמדת ועדת המחשבים הישראלית, גוף המייעץ למשרד המסחר והתעשיה, שרשתות מחשבים הוא אחד מבין ארבעה הנושאים הדחופים ביותר העומדים בפני משק המחשבים הישראלי.

* הכרת חשיבות הנושא ע"י הממונה על האוטומציה ומינהל מערכות באגף התקציבים, בזאת שהוקם לאחרונה גרעין של צוות קבוע באילתם לטפל בנושא של רשתות תקשורת מחשבים.

* החלטתה של המועצה העליונה להכוונת ענ"א עוד ב-1973, שיש מקום לבחון באופן קונקרטי את הנושא של רשתות מחשבים בארץ.

על הרקע הנ"ל, נראה כי התנאים בשלים לביצוע פרויקט צנוע מהסוג המוצע במסמך זה, אך מסקנה זו מותנת בהשגת המימון הדרוש לביצועו בצורה נאותה. אי אפשר לגשת לפרויקט כזה באופן חובבני; צריכה להתקבל החלטה ברורה וחד-משמעית בדרגי עיצוב מדיניות המחשבים בישראל, לפיה הולכים בכל הרצינות להקים ולבחון רשת מחשבים נסיונית ומגישים לצורך זה את מלוא הגיבוי והמשאבים הדרושים לכך. מה גם שמכירים בעובדה שהתועלת העיקרית לא תראה בשנה הראשונה או השניה לאחר הקמת הרשת, אלא בעתיד הרחוק יותר. אי-אפשר לשקול את הכדאיות הכלכלית של פרויקט מסוג זה בטווח הקצר, ולכן הצעה זו לא דנה כלל במונחים כאלה. כמו שבונים עיירות פיתוח לשם השגת יעדים לאומיים ארוכי-טווח, גם צריכים לבנות רשת מחשבים נסיונית לשם תרומתה לישראל בטווח הארוך.

ידועים היטב למחבר מסמך זה, הקשיים הכספיים בהם נתונה ישראל עכשיו, ומובן מאליו שמצבו של המשק לא ישתפר כה מהר. בתנאים כאלה פרויקטים מסוג זה הם בין הראשונים בדרך כלל להדחות. אך הנסיון מלמד שפיתות והחדרת השימוש בכל טכנולוגיה מתקדמת הוא תהליך אבולוציוני ומבחינה זו אין שוני בין רשתות מחשבים לבין כל טכנולוגיה אחרת, כולל מחשבים עצמם לפני 25 שנים. אם לא מוציאים את הסכומים הדרושים להתחלת התהליך כעת, פרושו של דבר דיחוי והזנחה נוספים, הפעם למועד בעתיד הרחוק. לכן אם הדרגים המוסמכים כבר הכירו בחשיבות הנושא, ואם שיתוף הפעולה מובטח מצד האוניברסיטאות, דומני שאין סיבה לדחות את הפרויקט פעם נוספת, אך חוזר אני ומדגיש שעדיף לדחות מאשר להתחיל "על רגל שמאל", ולא להעמיד את מיטב המשאבים - האנשים והאחרים - לרשות הפרויקט בכדי למנוע אכזבה, כשלון ודחייה ארוכה יותר.

2. יעוד הפרויקט ומטרותיו

באופן כללי ביותר יעודה של רשת מחשבים נסיונית בישראל, הוא לצבור ידע על הפעלת רשתות מחשבים ולמצוא פתרונות לבעיות הרבות הכרוכות בכך ע"י פיתוחה והפעלתה של רשת אב-טפוסית צנועה במסגרת מרכזי חישוב אוניברסיטאיים קיימים.

המטרות הספציפיות של הרשת הן :

(א) להעמיד לרשות הצרכנים של כל מרכז חישוב שמתחבר לרשת שרותים ומשאבים מירביים במחיר נמוך ככל האפשר ע"י :

* השלמת השרותים והמשאבים הקיימים באופן מקומי עם שירותים מיוחדים הקיימים רק במחשב אחר ברשת.

* הקטנת הכפילות הקיימת במערכות תכנה-בקרב האוניברסיטאות.

* הספקת שרותי תעוד וקיטלוג המאפשרים לצרכני הרשת למצוא ולהשתמש בשרותים המתאימים ביותר לפתרון בעיותיהם במחיר הנמוך שאפשר להשיג.

(ב) להעשיר את מגוון השירותים הניתנים לצרכני המחשבים החברים ברשת מעבר לאלה הקיימים כיום ע"י :

* שיתוף פעולה בכל הכרוך בנתוח דרישות הצרכנים לשרותי מחשב, איתור המוצרים המתאימים, והשגתם בדרך אופטימלית מתוכננת מראש מבחינת קהילת הרשת הנרחבת.

* לחלק משאבים ולקבוע סדרי עדיפויות בצורה גמישה ואופטימלית יותר בהשוואה למצב הקיים במסגרת מספר אירגונים נפרדים.

(ג) תכנון ורכישת ציוד נוסף באופן מרוכז ע"מ להשיג ניצולת טובה יותר מזאת שאפשרית ללא שיתוף פעולה בין המרכזים השונים. כך לדוגמה, ציוד נוסף לא יירכש, בדרך כלל, ע"י מרכז מחשבים ספציפי אם דרישותיו לתוספת כח מיחשוב יכולים למצוא פתרונן במסגרת המשאבים הקיימים ברשת.

(ד) לעודד שיתוף יתר בין האוניברסיטאות חברות ברשת בכל הקשור בפיתוח והחלפת מערכות, תכניות, מסדי נתונים ורעיונות לטובת כל צרכני הרשת.

(ה) לפתח התמחות בנושאים ספציפיים בכל מרכז חישוב החבר ברשת ע"מ להעמיד את מיטב השרותים לצרכנים תוך כדי ניצולת הסגל המקצועי של כל מרכז בצורה אופטימלית יותר.

דוגמאות אפשריות לשטחי התמחות הן :

* עיבוד נתונים מינהלי.

* מערכות ניהול ספריה.

* תכניות יישום השייכות לדיציפלינות אקדמיות שונות.

* מערכות הוראה בעזרת המחשב.

* מערכות לשליפת מידע טכני ומדעי.

(ו) להוות תשתית ל- communications-oriented computer utility לאומית עם הכוונה המוצהרת של הספקת שרותים נאותים לצרכן הישראלי (גם האקדמי וגם הלא-אקדמי), תוך כדי חסכון במטבע זר. מימוש מטרה זו בטווח הארוך, חייב להתייחס לזכויות הלגיטימיות של הגורמים המסחריים הפועלים בשוק לשרותי מחשב בארץ וגם צריך להיות מתואם עם המדיניות לפתוח ענף המחשבים כפי שתקבע ע"י הגופים המוסמכים לכך.

(ז) להוות מוקד למחקר שימושי במדעי המחשב בישראל ע"י :

* קיום מסגרת מחקרים קבועה לנושא רשתות מחשבים התואמת את יתר המטרות כפי שמוגדרים כאן.

* קיום נקודת משיכת חוקרים מחו"ל המאפשרת להם לתרום לפתוח הרשת הישראלית בפרט ולצבירת ידע בנושא הרשתות בכלל.

* קיום אפשרות לחוקרים ישראליים לבוא במגע ולשתף פעולה עם עמיתהם בחו"ל על מגוון האספקטים של פיתוח ויישום רשתות מחשבים.

(ח) לאפשר בחינת הגישה של רשת מחשבים כאמצעי הספקת שרותי מחשב בכל הסקטורים של המשק הישראלי כולל :

* ממשלה.

* בטחון.

* תעשיה.

* מסחר, ועוד.

(ט) להוות בסיס לשיתוף פעולה וחיבור עם רשתות מחשבים בחו"ל.

3. מטרות בטווח הקצר

בשלב הראשון של הפעלתה, הרשת תתרכז בהשגת מטרות א' ו-ב' כפי שהוגדרו לעיל. כדי להשיג מטרות אלה מבנה הרשת יהיה פשוט ככל האפשר ויאפשר שירותים כדלקמן לצרכני הרשת :

(א) גישה On-line לכל מערכת ותכנית הפועלת בשיטת "שיתוף הזמנים" בכל מחשב ברשת (או לחלק מהן לפי שיקול דעת מבצעי הפרויקט). לאחר שהמשתמש הרחוק יוצר קשר עם תכנית היישום בה הוא רוצה לעבוד, התקשרות והאינטראקציה יתבצעו באופן שקוף למשתמש.

(ב) שרותי Network Batch אשר יאפשרו לצרכן של מרכז חישוב חבר ברשת, להגיש "עבודת מכלול" (פקודות, תכנית ונתונים), לביצוע במחשב מרוחק ולכוון את תוצאות העיבוד למחשב המקומי שלו, או לכל מחשב או מסוף המחובר לרשת. על המשתמש בשרות זה לבנות את הפקודות, התכנית והנתונים שלו כפי שמחייב המחשב המטרה אשר יבצע את עבודתו.

(ג) שרותי העברת קבצים - אשר יאפשרו לצרכן הרשת להעביר קובץ סידרתי ממחשב למחשב ברשת. בכלל, יעמדו לרשות המשתמש המרוחק כל שרותי ניהול הקבצים של מחשב המטרה שלו, ועליו יהיה להשתמש בשרותים אלה כדי ליצור, לתחזק ולעדכן קבצים לאחר העברתם למחשב הרחוק.

(א) העצמאות של כל מרכז מחשבים חבר ברשת, תישמר ככל האפשר במשך שלב ההפעלה הראשונית. הכוונה היא לבחון את המטרות של הרשת (סעיפי 2 א' ו-ב' לעיל), ע"י הדגמה מרצון ולא ע"י אכיפה.

(ב) הנחה יסודית היא שהמנהל של כל מרכז מחשבים חבר הרשת, נושא גם באחריות וגם בסמכות להעמיד לרשות צרכניו את מיטב השרותים לפתרון בעיות המיחשוב שלהם. כוונת הרשת היא להשלים את השרותים המסופקים באופן מקומי ולפתוח בפני המרכזים דרכי הספקה חדשות ע"י שיתוף פעולה בין מרכזי המחשבים חברים ברשת. בשום מקרה לא יוחלף שרות מרחוק ללא הסכמת כל הגורמים הנוגעים בדבר.

(ג) במשך שלב ההפעלה הראשונית של הרשת, גישה למחשבים מרוחקים תינתן בדרך כלל לכל הצרכנים של מרכזי החישוב החברים ברשת. גישה לצרכנים אחרים וסירוב לאפשר גישה לצרכן מסויים יצטרכו לקבל אישור מיוחד מטעם מועצת המנהלים של הרשת.

(ד) כל גישה למחשבים רחוקים ברשת תהיה כפופה לתהליכים ולפרוצדורות המקובלים אצלם מבחינת פתיחת חשבונות, נוהלי עבודה וכו'.

(ה) כל צרכן מרוחק יקבל הקצבת אמצעים מאחד המרכזים ברשת לצורך רישום ובקרה ויקבל גם חשבונות תקופתיים בהתאם למקובל בכל מרכז חישובים. ההקצבות לצרכנים מרוחקים יקבעו ע"י כל מנהל בהתאם למאגר של משאבים וכללי חלוקתם אשר יוסכמו אחת לשלושה חודשים ע"י מועצת המנהלים של הרשת. כל צרכן יהנה בדרך כלל מסדר עדיפות שווה בכל הקשור עם גישה למחשב וביצוע עבודות. לא תהיה הבחנה בין צרכנים מקומיים וצרכנים מרוחקים. אין כוונה להעביר סכומים ממשיים בין מרכזי חישוב חברים ברשת במשך שלב ההפעלה הראשון, אך נקודה זו טעונה אישור שנתי ע"י מועצת המנהלים.

(ו) מענקים לצורך מימון הפרויקט יתבקשו מהגורמים המתאימים (הממשלה), אשר יהיה בהם כדי לכסות את הוצאות הקמת הרשת והפעלתה במשך שנתיים. כל מרכז חישוב חבר ברשת יספק כח-אדם אשר יטול חלק אקטיבי בפיתוח הרשת, וכן זמן מחשב. זמן המחשב יסופק ללא תמורה. תנאי זה נקבע בהתאם לעקרון שיש לדרוש התחייבות של ממש מכל מרכז חישובים בכדי להבטיח אינטרס בביצוע המוצלח של הפרויקט. בנוסף לאמור לעיל, כל מרכז חישוב יספק שרותי תעוד ויעוץ לכל משתמש מרוחק ברמה שווה לאלה הנתנים לצרכנים מקומיים. כמובן שכל מרכז מתחייב מראש להעמיד את כל תכניותיו לרשות כל צרכני הרשת.

(ז) פרויקט הרשת ינוהל ע"י מועצת מנהלים אשר תשמש גוף עיצוב מדיניות וקבלת החלטות עבור הפרויקט. אילתם תשמש הזרוע המבצעת של מועצת המנהלים ועליה תוטל אחראיות עבור תיאום הפיתוח והיישום של הרשת. המבנה האירגוני מוצג בשרטוט 1. מועצת המנהלים תורכב משני נציגים מוסמכים מכל מוסד החבר ברשת. יתווספו לחברים אלה נציג של הגורם המממן את הפרויקט וכן נציג מאילתם. מועצת המנהלים רשאית להזמין לפגישותיה חברים "בתוקף תפקיד" במטרה להתייעץ איתם ולתאם פעולות הרשת עם גורמים אחרים.

(ח) פעולותיה היום-יומיות של הרשת יבוצעו באמצעות חמישה צוותות כמפורט להלן :

* צוות שרותי רשת ותיעוד - אשר יהיה אחראי על קיום מגע עם צרכני הרשת בכל הנוגע לשרותי הרשת. צוות זה יורכב מראשי קבוצות היעוץ של כל מרכז חישובים חבר הרשת.

* צוות המינהלה - אשר יהיה אחראי על סידורי גישה והקצבת משאבי מחשב ובנוסף ענינים מינהליים אחרים, כגון סידרי עדיפויות וכו'. צוות זה יורכב מנציגים מינהליים מכל מרכז חבר ברשת.

* צוות טכני - אשר יהיה אחראי על תכנון, תכנות ואחזקת התכניות במחשבים אשר דרושים לספק את שרותי הרשת. צוות זה יורכב ממתכנתי המערכות הראשיים של כל מרכז חבר ברשת.

* צוות תכנון וניהול הפרויקט - אשר יוקם באילתם ויהיה אחראי לפיתוח וביצוע הפרויקט והכוונתו.

* צוות תקשורת - אשר יהיה אחראי על תכנון, תכנות, אחזקה והפעלת המחשב הזעיר, אשר ישמש מחשב התקשורת של הרשת. תפקיד זה יוטל על אחד ממרכזי המחשבים החבר ברשת.

(ט) אילתם תמלא את התפקיד של הזרוע המבצעת של הפרויקט. מנהל הפרויקט באילתם ישמש מזכיר מועצת המנהלים. אילתם תחזיק צוות טכני צנוע (בין 2 - 3 אנשים), אשר מתפקידו לספק שרותי תכנון, עיצוב וניהול הפרויקט, כאשר הביצוע עצמו יתבצע בדרך כלל ע"י גורמים חיצוניים, הכל לפי מדיניות אשר תועבר למועצת המנהלים לאישור. אילתם תשמש כסוכן הבלעדי של הרשת בכל הנוגע לקיום מגע עם ספקים וגורמים אחרים בכל ענין מרכזי ועקרוני לרשת, אך פעולותיה והמלצותיה יהיו תחת פיקוח שוטף של מועצת המנהלים. באופן כללי כל מרכז חישובים חבר ברשת ישא באחריות היישום של תכניות הגישה והבקרה של הרשת לפי סטנדרטים אשר יקבעו ע"י הצוות של אילתם בתאום עם הצוות הטכני ויאושרו ע"י מועצת המנהלים.

(י) מוצע שמועצת המנהלים תקבע מדיניות להשגת טכנולוגיה אשר תעדיף אימוץ גישות קיימות ורכישת פריטים קיימים על פני מחקר ופיתוח מקוריים. מדיניות מסוג זה תואמת את יעדי הרשת כפי שפורטו לעיל. לפיכך מוצע לאמץ את סדר העדיפויות הרשום להלן כפוף כמובן לקריטריונים כלכליים, טכניים ואחרים :

* רכישת טכנולוגיה מוכנה ("מהמדף") בצורת מוצרי חמרה ותכנה קיימים או ידע ממוסדות וגורמים אשר מציעים למכירה טכנולוגיה שהוכיחה את עצמה בחו"ל.

* פיתוח טכנולוגיה או התאמתה ע"י אוניברסיטאות חברות ברשת.

* פיתוח טכנולוגיה או התאמתה ע"י מוסדות חינוך או מחקר ישראליים שאינם חברים ברשת או גורמים מסחריים ישראליים.

* פיתוח טכנולוגיה או התאמתה ע"י גורמים מסחריים זרים.

(יא) תפקידו העיקרי של אילתם במשך שלב ההפעלה הראשון, יהיה מעקב אחרי ביצוע הרשת והשימוש בה, מדידות וניתוחי הפעלתה, וקידום המטרות שפורטו לעיל. הפעלתה של הרשת באופן יום-יומי תוטל על אחת האוניברסיטאות חברות ברשת, אולי בשיתוף גורם מסחרי חיצוני.

5. לוח זמנים ואומדן תקציבי

(א) לצורך הערכת אומדני תקציב לפרויקט הרשת, הונחו ההנחות הבאות :

* בשלב ראשון הרשת תורכב מארבע אוניברסיטאות מבין שש הבאות בחשבון. רק לצורך עריכת אומדני התקציב, נלקחו בחשבון המוסדות הבאים: מכון ויצמן למדע, הטכניון, אוניברסיטת תל-אביב והאוניברסיטה העברית.

* המבנה של הרשת שנלקח בחשבון הוא מבנה כוכב (ראה שרטוט 2), עם מחשב זעיר אחד שיספק את שרותי המיתוג והנתוב הדרושים.

* קוי נל"ן רגילים ירכיבו את אמצעי התקשורת היסודיים עבור הרשת.

* טכנולוגיה קיימת תרכש בצורת מיפרטים, תכניות מחשב, וידע (שרותי יעוץ) מחו"ל. היישום יתבצע בארץ ע"י התאמת הטכנולוגיה המיובאת ותכנות מחדש עבור הרשת הנסיונית.

* מיסוי מלא מובא בחשבון על כל פריט אשר יירכש לצורך הקמת הרשת, למרות שהמדובר הוא בפרויקט במסגרת אקדמית אשר עשוי להנות מהקלות במיסים. שער החליפין המובא בחשבון הוא 8.00 ל״י לדולר, כך שיש להתאים את הערכות התקציב לשינויים בשער החליפין, וכמו כן לשינויים במערכת המיסוי.

(ב) פרויקט הרשת מחולק לשלושה שלבים במפורט להלן :

* שלב התארגנות ותכנון - 6 חודשים.

* שלב הקמה - 12 חודשים.

* שלב הפעלה ראשון - 24 חודשים.

(ג) אומדני תקציב מוצגים להלן לכל שלב בנפרד :

* שלב התארגנות ותכנון - מאמץ הדורש תעסוקתם של שני אנשים, מבוצע ע"י אילתם ומיועד לייצר תכנית עבודה עבור הפרויקט, הכולל פרטים על הגישה הטכנולוגית. הגדרה מדוייקת יותר של שרותי הרשת לצרכניה, ואומדני תקציב מבוססים יותר, יסוד מרכזי לביצוע שלב זה של הפרויקט הוא השתלמות בת שלושה חודשים בחו"ל של מנהל הפרויקט באילתם, אשר תכלול לימוד מקרוב של מספר רשתות מחשבים נבחרות. מטרות ההשתלמות הן ללמוד לעומק את תולדות, דרך היישום, המבנה, סידורים מינהליים ואירגוניים, סידורי הפעלה, ולקחים של הרשתות בחו"ל.

שלב זה של הפרויקט ידרוש מימון כדלקמן :

- 12 חדשי-אדם	150,000 ל״י
- הוצאות נסיעה (אש"ל והתחבורה לשלושה חדשים)	50,000 ל״י
סה"כ שלב ראשון	200,000 ל״י

* .<u>שלב ההקמה</u> – מאמץ ה- שנה אשר מטרתו ביצוע התכנון המפורט, רכישת הציוד הדרוש, וכתיבת התכניות הדרושות להפעלת הרשת. במשך שלב זה יתבצעו משימות אחרות כגון : הכנת מדריכים למשתמשים, פיתוח קורסים למשתמשים, קביעת תהליכים מינהליים וכו'. אומדן תקציבי לשלב זה של הפרויקט מתבסס על הקונפיגורציה המוצגת בשרטוט 2 ועל היקף של 8 שנות אדם לפי התפלגות הבאה :

- ניהול הפרויקט (אילתם) – 2
- פיתוח תכניות למחשבי הרשת – 4
- תכניות למחשב הזעיר – 1
- משימות אחרות – 1

בנוסף לכח האדם הנ"ל מוקצב סכום של 50,000 $ עבור שרותי יעוץ אשר יירכשו בחו"ל. סה"כ האומדן לביצוע שלב ההקמה מפורט להלן :

1. <u>חמרה</u> :			
– מחשב זעיר	30,000	$	
– מודמים	70,000	$	
	100,000	$ =	
	800,000	₪	
– מיסים (50%)	400,000		
– הוצאות יבוא	50,000		
סה"כ חמרה			₪ 1,250,000
2. <u>יעוץ</u> :			
– שרותי יעוץ מחו"ל	50,000	$ =	
	400,000	₪	
– מיסים (15%)	60,000		
סה"כ יעוץ			₪ 460,000
3. <u>כח-אדם</u> :			
– 8 שנות אדם לפי ההתפלגות הנ"ל			₪ 1,200,000
4. <u>הוצאות שונות ובלתי מתוכננות מראש</u> :			₪ 90,000
סה"כ ביצוע שלב ההקמה			* ₪ 3,000,000

---

* זמן מחשב חינם במחשבים חברי הרשת

* שלב ההפעלה הראשון – במשך שלב זה הרשת תפעל. סעיפי ההוצאות העיקריים יהיו שכירת קוי תקשורת, אחזקת הכנה, מדידת וניתוח ביצועי הרשת, הרחבת מספר בחברים ברשת והפעלת תת-רשת התקשורת. אומדני התקציב המובאים להלן מבוססים על צוות בן שלושה אנשים באילתם, תכניתן מערכת בכ"א ממרכזי החישוב חברי הרשת, תכניתן אחד למחשב הזעיר, וזמן מחשב חינם במרכזי המחשב חברים ברשת עבור מדידות וניתוחים הקשורים לרשת ושיפורים לתכנת הרשת.

- התקציב השנתי הדרוש לכיסוי הוצאות ההפעלה מוערכים כדלקמן :

– כח אדם	1,200,000 ל"י
– קוי תקשורת	150,000 ל"י
– ציוד תקשורת נוסף ($ 50,000 ו-50% מיסים)	600,000 ל"י
– תעוד	100,000 ל"י
– שונות ובלתי מתוכננות מראש	150,000 ל"י
סה"כ הפעלה שנתית	2,200,000 ל"י

* סה"כ הוצאות מתוכננות – לסיכומו של דבר, אומדן התקציב לכל שלבי הפרויקט מוערך כדלקמן :

שנה תקציבית	שלב	תקציב
1976/77	התארגנות ותכנון	200,000 ל"י
1977/78	הקמה	3,000,000 ל"י
1978/79	הפעלה ראשונית	2,200,000 ל"י
1979/80	הפעלה ראשונית	2,200,000 ל"י
		7,600,000 ל"י

6. **התיחסות גורמים ממשלתיים ואוניברסיטאים להצעה זו.**

לאחרונה התקיים דיון על הצעה זו בו השתתפו מנהלי מרכזי החישוב של שלוש האוניברסיטאות המפעילות מחשבים מתוצרת קדנטרול דטה (האוניברסיטה העברית, אוניברסיטת תל-אביב, ואוניברסיטת בן-גוריון) וכן נציג היחידה לאוטומציה באגף התקציבים ונציג אילתם.

במהלך הדיון הביעו נציגי האוניברסיטאות רצונם לקשור בין שלושת המוסדות על מנת להשיג שלוש מטרות עיקריות:

- חסכון בפיתוח רכישת מוצרי תכנה
- איזון עומס על המחשבים ע"י וויסות ביצוע עבודות
- קידום טכנולוגי בנושא רשתות מחשבים

נציג אגף התקציבים הביע את רצונו לקדם את נושא הרשתות בארץ ולשאת בחלק מהוצאות פרויקט בנושא זה בתנאי שהשותפים יהיו מוכנים להשתתף באופן פעיל ולתרום משאבים.

נוצרה שפה משותפת בדיון זה והוחלט להקים פרויקט בחסות ארבעת הגורמים הנ"ל. מחבר מסמך זה נתמנה במרכז פרויקט.

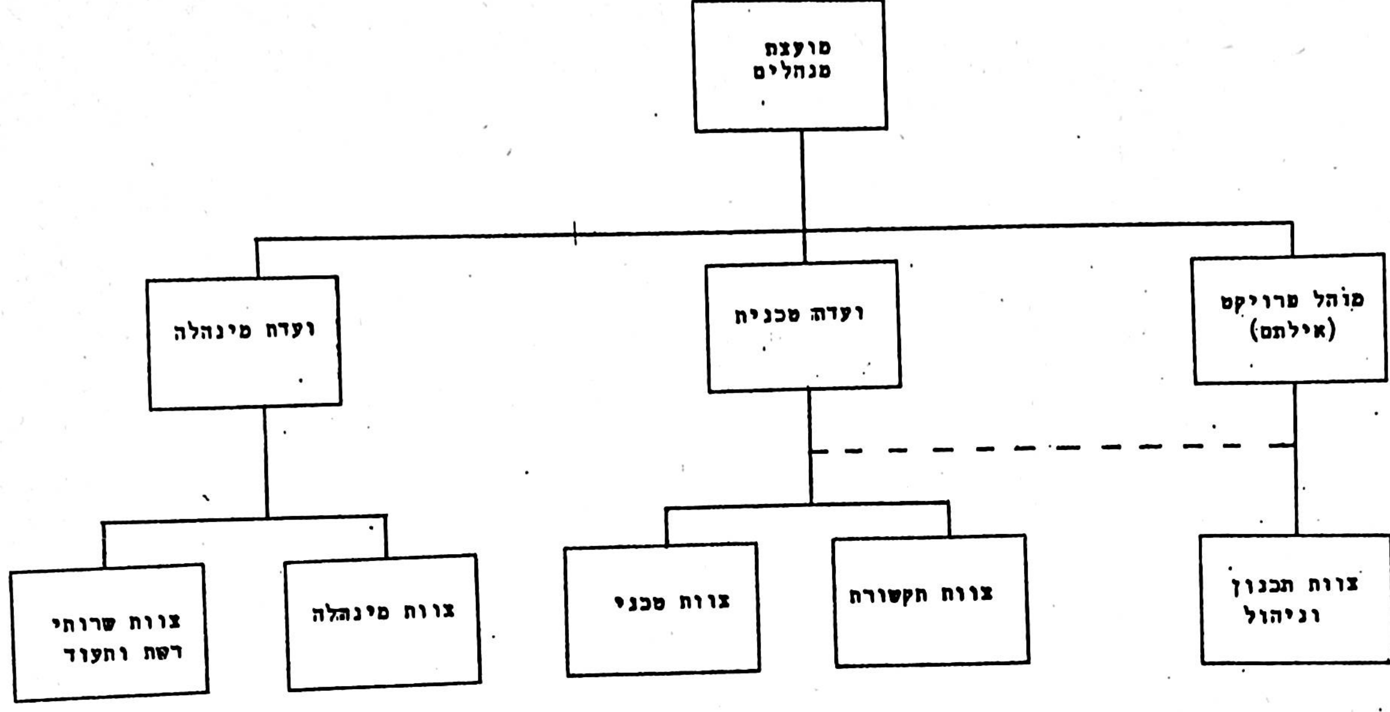
הצעד הראשון הוא לערוך בדיקת יישימות אשר תבוצע בשלושה שלבים כמפורט להלן.

א) לימוד שלושת מרכזי החישוב המשתתפים בפרויקט מבחינת הציוד והתכנה העומדים לרשותם, דפוסי השימוש של הצרכנים העיקריים שלהם, והאפשרויות לשתף פעולה ביניהם. יוכן דו"ח ביניים אשר יכסה את השלב הראשון ויכלול אומדני תקציב להקמת הרשת.

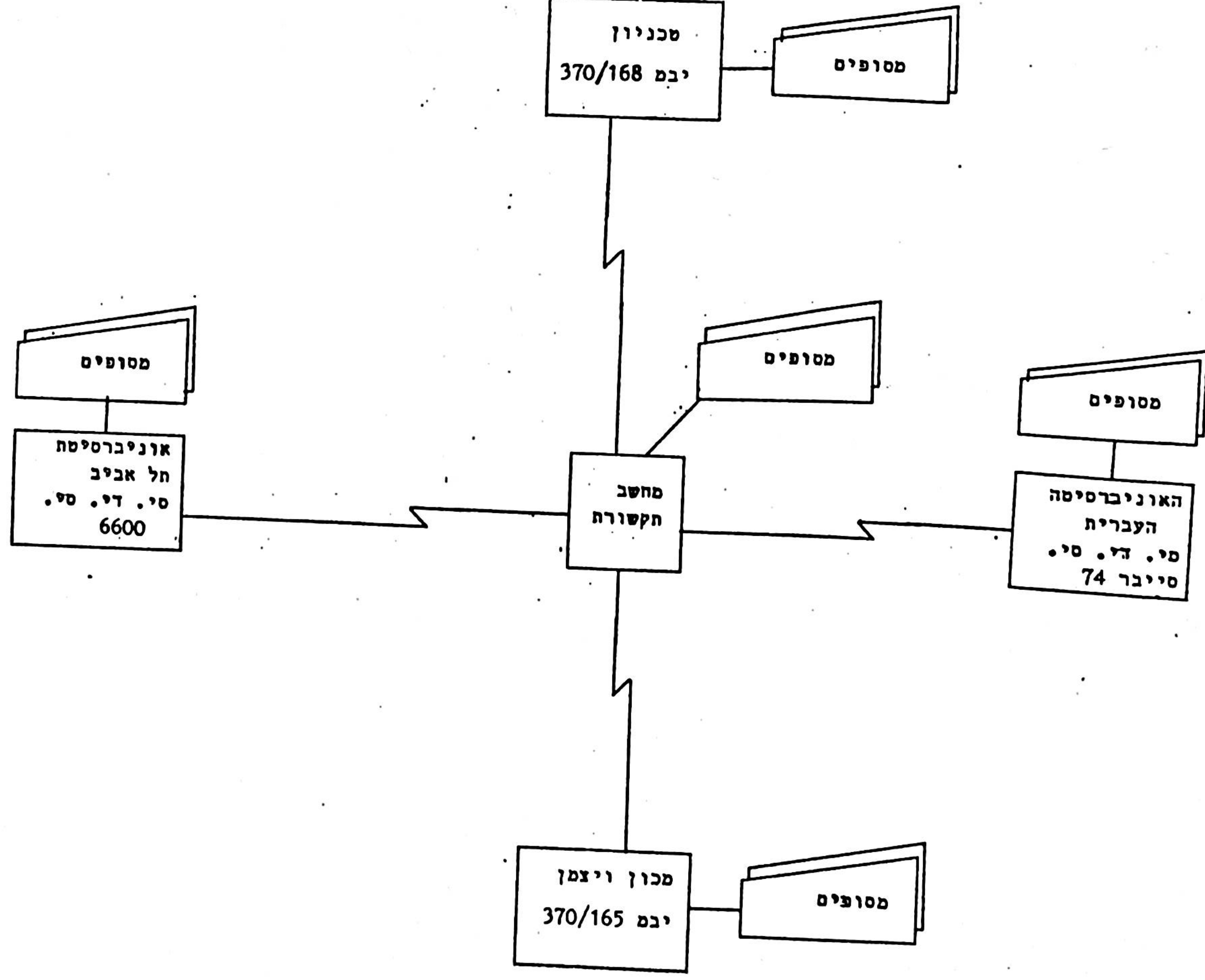
ב) דיון בועדה המנחה את הפרויקט ובהנהלת המוסדות המשתתפים על כיווני הפיתוח הרצויים והאפשריים, ועל קביעת התקציב לשלב הבא.

ג) ביקור בחו"ל אצל רשתות מחשבים נבחרות על מנת לעמוד מקרוב על הנסיון שהצטבר אצלן בהקמתה ובהפעלתה של רשת מחשבים ובהשפעה של הרשת על המוסדות שהיקמו אותה. לאחר ביקור זה יוכן דו"ח מסכם אשר יכלול תכנון כללי של הרשת, הערכות תקציביות, וציפיות מהרשת לאחר הקמתה, דו"ח זה יעמוד לדיון והחלטה על המשך הפרויקט.

לא יהיה נכון לשער שקשר בין שלוש האוניברסיטאות הנ"ל הוא המטרה הסופית של הפרויקט הזה. אלא מוסדות אלה מהווים בסיס נוח, הן מבחינה טכנית והן מבחינה אירגונית, לביצוע עבודה מעשית לכיוון הקמת רשת מחשבים ישראלית. לא מן הנמנע שמוסדות אחרים יצטרפו לרשת בעתיד וכמו-כן רצוי שהרשת הישראלית תתחבר לרשתות זרות בעת המתאים. לכן יש לראות בצעדים אלה נסיון להעלות את מדינת ישראל על מפת רשתות המחשבים העולמי בצורה ובקצב המתאימים.



שרטוט 1 – מבנה אירגוני


טכניון
יבם 370/168
מסופים
אוניברסיטת
תל אביב
סי. די. סי.
6600
מסופים
מחשב
תקשורת
מסופים
האוניברסיטה
העברית
סי. די. סי.
סייבר 74
מסופים
מכון ויצמן
יבם 370/165
מסופים

קבוצת דיון מס' 9:

"יישומים של מדעי המחשב התאורטיים"

יו"ר: פרופ' י. כצנלסון

ש. כץ

א. בונה

מ. רודה

אימות אוטומטי של תוכניות

שמואל כ"ץ
מרכז מדעי יבם ישראל

מבוא.

מתכנת הכותב תוכנית למחשב חייב לבדוק את תוכניתו ולנקותה משגיאות. תחילה הוא מתקן הפרות של כללי שפת התכנות (כגון אי הכרזה על משתנה או מערך, משפט לא חוקי וכוי), תוך שמוש בהודעות התוכנית המתרגמת. לאחר מכן עליו לנפות שגיאות לוגיות בתוכנית. בעיה זאת קשה יותר לפתרון. אפשר להעזר, במקרה הטוב, ב- trace של בצוע התוכנית עם הקלט המסוים. כלומר, אפשר לקבל את ערכי המשתנים בנקודות מסוימות של התוכנית לקלט נתון.

בהמשך תתוארנה בקוים כלליים שתי טכניקות שפותחו במרכזי מחקר שונים בעולם במטרה לוודא שאמנם תוכנית מחשב "נכונה" - כלומר, שאין בה שגיאות. נדגיש שמדובר כאן בשגיאות לוגיות. ההנחה היא שהתוכנית חוקית בשפה וברצוננו לדעת אם היא פועלת כהלכה. בסוף אנסה לנתח את ההשפעה המעשית שתהיה לשיטות אלו בעתיד.

עצה לקורא.

מאמר זה מהווה סקירה. הוא איננו מספיק על מנת להבין לעומק את השיטות המובאות. לקורא המתעניין מומלץ ספרו של פרופי זהר מנה [1974] Manna כמבוא לשיטות הוכחה (פרק 3). חומר נוסף מצוי במאמרים המופיעים במקורות.

כדאי לציין שבטכניון, במכון ויצמן ובאוניברסיטת תל-אביב וגם באוניברסיטאות אחרות ניתנים קורסים בנושא.

ספציפיקציה ונכונות.

מהי תוכנית "נכונה"?
תחילה עלינו לרשום באופן מלא ומדויק מהן הספציפיקציות של התוכנית (הדרישות ממנה). מפליא כמה קשה, לעתים, לעשות זאת. בתוכנית מיון של וקטור, למשל, רוצים שהאיברים בוקטור יהיו בסדר עולה בסוף התוכנית, אבל אסור לשכוח שהם חייבים להיות תמורה של הוקטור המקורי.

לתוכניות שמחשבות פונקציה כלשהי של הקלט (להלן "תוכניות פונקציונאליות") ניתן לרשום את הספציפיקציה בשני חלקים:

(א) דרישות על הקלט-כלומר, הגדרת הערכים האפשריים של הקלט;

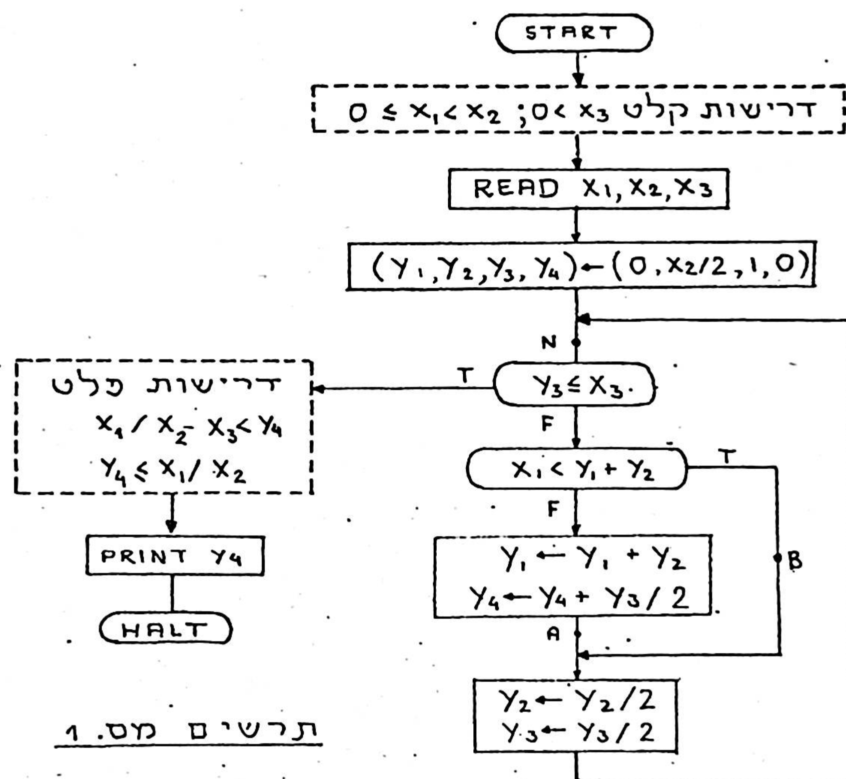
(ב) דרישות על הפלט-כלומר, היחסים הרצויים בין הקלט והפלט.

את הספציפיקציה מקובל לרשום בשפה של לוגיקה מתמטית - תחשיב הפרדיקטים מסדר ראשון. ניתן גם לרשום ספציפיקציה בצורה יותר טבעית, והתפרסמו כמה עבודות על פתוח שפה מתאימה לביטוי טענות כאלו.

תוכניות מסחריות רבות שתפקידן לשנות ולסדר מבני נתונים, הינן תוכניות פונקציונאליות, ואולם קיימות תוכניות רבות שאינן פונקציונאליות. מערכת הפעלה, למשל, אינה ניתנת לתיאור כפונקציה פשוטה של הקלט (שהוא במקרה זה סדרה של תוכניות משתמשים). הרחבת מושגי הספציפיקציה כך שיתאפשר טפול גם בתוכניות בעיתיות כאלו הוא נושא למחקר.

על תוכנית (פונקציונאלית) נאמר שהיא נכונה לגבי ספציפיקציה נתונה אם ורק אם לכל קלט המקיים את דרישות הקלט (להלן "קלט חוקי") התוכנית עוצרת עם פלט הממלא את דרישות הפלט. מכאן שתוכנית תקרא לא נכונה לגבי ספציפיקציה נתונה אם ורק אם קיים לפחות קלט חוקי אחד שעבורו התוכנית אינה עוצרת (בגלל ביצוע אינסופי של לולאה), או יוצרת פלט שגוי(שאינו מקיים את הדרישות).

דוגמא 1:



תרשים מס. 1

תרשים מס' 1 מתאר תוכנית בה הוספנו ספציפיקצית קלט לפני פקודת read וספציפיקצית פלט לפני פקודת print. שים לב לחשיבות הדרישות על הקלט: בתוכנית זאת מצפים לקבל אך ורק ערכים ממשיים המקימים $0\leq X_1<X_2$ ו- $0<X_3$.

לכל קלט אחר התנהגות התוכנית אינה מעניינת. עבור קלט חוקי התוכנית צריכה לחשב ב- Y_4 את X_1/X_2 בקירוב של X_3 וזאת על ידי חבורים וחילוק ב-2 (הניתן למימוש כ- shift right).

בדיקה או הוכחה.

איך אפשר להראות שתוכנית אמנם נכונה לגבי ספציפיקציה נתונה? בדרך כלל, לא ניתן לנסות כל קלט אפשרי מפני שמספר האפשרויות הוא עצום. הדרך המקובלת היום היא לבדוק את התוכנית על ידי מספר קטן של ריצות עם נתוני קלט שונים (שיטת ה- testing). בשיטה זאת אין בטחון שהתוכנית נכונה עבור קלט שלא נבדק (ובמיוחד עבור מקרים "מנוונים" של נתונים). אפילו אם בבדיקה מסתבר שיש שגיאה בתוכנית, קשה לעתים קרובות לאתר אותה.

בהמשך נציג שתי שיטות שנועדו להתגבר על החסרונות הנ"ל:

(א) בצוע סמבולי - שמטרתו להבטיח שיטתיות בהרצות התוכנית על סוגי קלט שונים.

(ב) הוכחת נכונות - באופן מתמטי, מבלי להריץ את התוכנית כלל.

בצוע סימבולי.

במקום לבצע testing עם קלט שנבחר באקראי, נגדיר קבוצות של נתוני קלט שהפעלת התוכנית עליהם עוברת במסלולים אותם אנו רוצים לבדוק, באופן הבא: במקום ערכים ממשיים, נותנים לכל משתנה קלט סימן כגון- $C_1, C_2, \ldots$ ומבצעים סימולציה של פעולת התוכנית על גבי הערכים הסימבוליים במקום על ערכים מספריים. המטרה היא ליצור את התנאים כדי "ללכת" על המסלולים השונים מהתחלת התוכנית.

נדגים את הרעיון באמצעות דוגמא 1. נתחיל ב-

$$X_1=C_1 \wedge X_2=C_2 \wedge X_3=C_3.$$

(הסימן "∧" משמעותו - "וגם". $P\wedge Q$ מסמן ש-P נכון וגם Q נכון).

ברור ש- $0<C_3 \wedge 0\leq C_1<C_2$ בהתחלת התוכנית. אחרי שמגיעים לנקודה N בפעם הראשונה ברור גם ש-

$$Y_1=0 \wedge Y_2=C_2/2 \wedge Y_3=1 \wedge Y_4=0.$$

כדי לצאת מהבדיקה הראשונה ביציאה T דרוש ש- $Y_3\leq X_3$, כלומר ש- $1\leq C_3$, מכאן שקלט שעבורו לא תתבצע הלולאה אף פעם מקיים:

$$0\le C_1\le C_2 \wedge 0<C_3 \wedge 1\le C_3$$

נניח שברצוננו למצוא תנאי לכך שהלולאה תתבצע פעם אחת, דרך B: מהיציאה F של $Y_3\le X_3$, ברור ש- $1>C_3$. מהבדיקה השניה $C_1<0+C_2/2$. במסלול הזה Y_3 הופך להיות 1/2, לכן התנאי ליציאה ל- $HALT$ הוא $1/2\le C_3$. נקבל שהתנאי להליכה על מסלול זה הוא:

$$0\le C_1<C_2 \wedge 0<C_3 \wedge 1>C_3 \wedge C_1<C_2/2 \wedge 1/2\le C_3$$

לאחר פישוט נקבל:

$$1/2\le C_3<1 \wedge 0\le C_1<C_2/2$$

באופן דומה ניתן לבנות את התנאים המתאימים לכל מסלול בו אנו מעונינים. במקרה שלנו יש שתי דרכים לעבור מ-N חזרה ל-N (דרך A או B. לכן נקבל "עץ בינארי" של תנאים אפשריים - 4 לשני מעברים בלולאה, 8 לשלושה מעברים וכו'. ברור שאי-אפשר לבנות את העץ המתאים לכל האפשרויות, אבל ניתן לעשות זאת עבור "עומק" סביר.

בתנאים המתקבלים ניתן להשתמש בשתי צורות:

(א) לפתור את מערכת אי-השוויונות המבטאת את התנאי, לקבל דוגמא של נתוני קלט הממלאים את כל הדרישות ולהריץ את הפתרונות כ- Test. אם התוכנית נכונה עבור כל נתוני הקלט האלו, גדלה מידת הבטחון שבדקנו היטב את התוכנית. ואולם אין בכך עדיין הוכחה שתמיד תעבוד התוכנית כהלכה.

(ב) להוכיח את נכונות המסלולים (במידה שיש לתוכנית ספציפיקציה). כלומר, להראות שלכל קלט שימלא את התנאים למסלול, התוכנית נכונה. מובן שלקלט כזה התוכנית מסתיימת. אם, בנוסף, הספציפיקציה ניתנת להוכחה מתוך מה שידוע על המשתנים, המסלול תקין. לדוגמא, במסלול $HALT\text{-}N\text{-}START$, ידוע ש-

$$0\le C_1\le C_2 \wedge 1\le C_3 \wedge X_3=C_3 \; X_2=C_2 \wedge X_1=C_1 \wedge Y_4=0$$

בתנאים אלו ברור ש-

$$(X_1/X_2-X_3)<Y_4\le X_1/X_2$$

כי

$$(C_1/C_2-C_3)\le(C_1/C_2-1)<0\le C_1/C_2$$

במקרה של כישלון בבדיקה מסוג (א) או בהוכחה של מסלול (שיטה (ב)), מקבלים ידע רב, באופן יחסי, על נתוני הקלט הבעיתיים ועל הקטע בתוכנית

הגורם לטעות.

עבודה רבה נעשתה בפתוח מערכות אוטומטיות לבצוע סימבולי של תוכניות, ביניהן על ידי [1974] King ו-[1975]Boyer, et al. המערכות עדיין נסיוניות, אך התוצאות טובות בשתי הדרכים שתוארו.

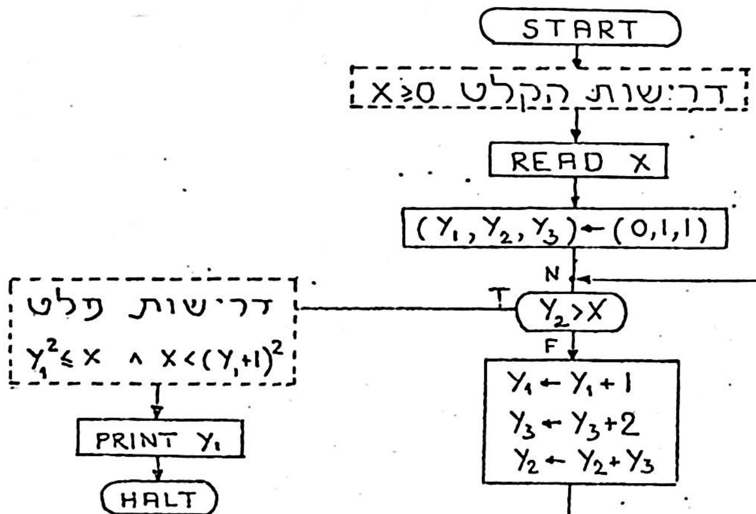
הוכחה של נכונות חלקית וסיום - שיטת [1967] Floyd.

בהוכחת נכונות של מסלולים לא ניתן "לסגור" את ההוכחה כך שתטפל בכל התוכנית, כיוון שעשויים להיות אינסוף מסלולים. כדי לקבל הוכחה מלאה, נצטרך להשתמש באינדוקציה. כאן נראה שיטה המבוססת על אינדוקציה על החשוב עצמו. "חותכים" את כל הלולאות בתוכנית כך שמנקודת חתך אחת לשכנתה יש מספר מסלולים פשוטים שאינם עוברים דרך נקודות חתך, ומוכיחים שאם החשוב "נכון" כשמגיעים לנקודה A, אזי הוא נכון בכל נקודות החתך הסמוכות ל-A.

כמו בכל הוכחה באינדוקציה, דרושות הנחות אינדוקציה. אלה הן טענות שמצרפים לנקודות החתך. נוכיח לכל מסלול פשוט שאם ההנחה בנקודת ההתחלה של המסלול היא נכונה, ואם נבצע את המסלול, אזי ההנחה בסוף המסלול תהיה נכונה עבור הערכים החדשים של המשתנים. אם נבחר את ההנחות בצורה מתאימה (ואם התוכנית באמת נכונה) נצליח להראות שדרישות הפלט נובעות מהנחות האינדוקציה באותו מקום.

שים לב ששיטה זו אינה מבטיחה שנגיע אי פעם לנקודת עצירה. אנחנו מוכיחים רק נכונות חלקית, כלומר - אם בבצוע מגיעים לנקודת עצירה, אזי תוצאות התוכנית נכונות. בהוכחת סיום צריך לטפל בנפרד. לנקודה זאת נתיחס בקצרה בהמשך.

דוגמא 2:



תרשים מס 2.

הבה נתבונן בתרשים מס. 2. בתוכנית שלושה מסלולים פשוטים: מ- *START* ל- *N*, מ- *N* סביב הלולאה חזרה ל-*N* ומ-*N* ל- *HALT*. כהנחות אינדוקציה נקח את ספציפיקציות הקלט והפלט, ובנקודה *N* את הטענה:

$$.Y_3=2Y_1+1 \wedge Y_2=(Y_1+1)^2 \wedge Y_1^2 \le X$$

לגבי המסלול מ- *START* ל- *N*: ידוע שבתחילה $0\le X$ ולאחר בצוע הקטע:

$$.\ Y_3=1 \wedge Y_2=1 \wedge Y_1=0$$

לפיכך כשמגיעים ל- *N* בפעם הראשונה,

$$1=2\cdot0+1 \wedge 1=(0+1)^2 \wedge 0\le X$$

כעת נניח שהנחת האינדוקציה נכונה ב- *N*, ושהתוכנית עוברת על המסלול מ-*N* ל- *N*. נרצה להוכיח את הנחת האינדוקציה עם הערכים החדשים בסוף המסלול. זאת אומרת אנו מניחים

$$Y_2\le X \wedge Y_3=2Y_1+1 \wedge Y_2=(Y_1+1)^2 \wedge Y_1^2\le X$$

אם נסמן ב- Y_1', Y_2' ו- Y_3' את הערכים בסוף המסלול, אזי

$$Y_3+2=Y_3' \wedge Y_2+Y_3+2=Y_2' \wedge Y_1+1=Y_1'$$

באשר Y_1, Y_2 ו- Y_3 הם הערכים בהתחלת המסלול. לכן אנו רוצים שיתקיים

$$Y_3+2=2(Y_1+1)+1 \wedge Y_2+Y_3+2=((Y_1+1)+1)^2 \wedge (Y_1+1)^2\le X$$

ואכן, לאחר בצוע מספר פישוטים רואים שהיחסים אכן מתקיימים.

לגבי המסלול מ- *N* ל- *HALT* ההנחות הן

$$.X<Y_2 \wedge Y_3=2Y_1+1 \wedge Y_2=(Y_1+1)^2 \wedge Y_1^2\le X$$

לכן, ב- *HALT*

$$.Y_1^2\le X \wedge X<(Y_1+1)^2$$

אלו הן בדיוק דרישות הפלט, ובכך הוכחה נכונות חלקית. כדי להוכיח שהתוכנית מסתיימת ננצל את העובדה ש- Y_1 מונה את מספר הפעמים שהלולאה מתבצעת ושמתקיים $Y_1^2\le X$ בכל פעם שמגיעים ל- *N*. ברור שאי אפשר לבצע את הלולאה יותר מ- $\lfloor\sqrt{X}\rfloor$ פעמים כי אז Y_1 יסתור את מה שהוכחנו, ולכן זה מוכיח סיום לכל קלט חוקי.

בתוכניות אחרות נוח לעתים להוסיף משתנה "מדומה" שמונה את מספר הפעמים שכל לולאה מתבצעת, ולהוכיח שהמונה חסום (אבסולוטית) מלמעלה בנקודת החתך של הלולאה.

מערכות רבות פותחו לשם הוכחת תוכניות לא גדולות בשיטה שתוארה;

.(Waldinger and Levitt [1974],Deutsch [1973])

הבעיה הקשה יותר של מציאת הנחות האינדוקציה אף היא זוכה למחקר ואוטומציה בשנים האחרונות.

המצב הנוכחי

בנוסף למוזכר למעלה, מצויות כיום כמה מערכות המטפלות במיוחד בתוכניות רקורסיביות. כמו כן מתבצעת עבודה תיאורטית רבה לשם טיפול בתוכניות "אמיתיות" ובפרט: בעיות של העברת פרמטרים, מבני נתונים מסובכים וכו'. קיימות גם שיטות הוכחה שונות מזו של Floyd, למשל - אינדוקציה על ערכי הקלט האפשריים ולא, כפי שנעשה כאן, על החישוב.

הדבר המעכב פיתוח מערכות מעשיות הוא הצורך ב"מוכיח משפטים" אוטומטי שיראה קשר בין הנחות נתונות לטענות שיש לנו ענין בהן. מצויות אמנם שיטות לעשות זאת, אבל הן איטיות ודורשות מקום רב בזכרון. בשטח הזה דרוש שיפור ביעילות כי הוא מהווה את צוואר הבקבוק וזאת מבלי למעט בבעיות האחרות כגון מתן הספציפיקציות ומציאת הנחות האינדוקציה.

השלכות אפשריות

ההתפתחות הנראית לי סבירה היא בשלושה כוונים:

(א) שיפור בכתיבה ודוקומנטציה של תוכניות. תכנות במבנים (structured programming) מושפע ביותר מטכניקות אימות. אחת מאבני היסוד של תכנות במבנים היא החלוקה לתת תוכניות רבות כשכל אחת מהן קצרה, ניתנת לבדיקה קלה בנפרד, ומכילה הגדרה מדויקת של דרישות הקלט ודרישות הפלט. הדרישות האלה הופכות לחלק בלתי נפרד של תהליך כתיבת התוכנית. כמו כן, הראיון לחשוב על הנחות האינדוקציה במקביל או אפילו לפני כתיבת התוכנית מקל על כל נסיון להוכיח נכונות; ומשפר את ההבנה של תהליך כתיבת תוכניות. קורסי תכנות בעתיד ידגישו יותר ויותר עקרונות אלו, תוך שילוב ידע על אימות.

(ב) למערכות אוטומטיות לבדיקה על ידי ביצוע סימבולי צפוי ישום מעשי בשנים הקרובות. הכלים הדרושים לכך קיימים ויעילותם סבירה בהשוואה לזמן הדרוש לנפוי שגיאות בשיטות המקובלות.

(ג) סביר שתכתבנה מערכות לאבחון של טעויות לוגיות בדומה לאבחון טעויות בשפה, וזאת על בסיס שיטות להוכחה (לאו דוקא זו שהודגמה למעלה). אין לצפות למערכת לגמרי אוטומטית, אלא לדו-שיח בו המשתמש מכוון ומיעץ למערכת

בנסיון להוכיח או לתקן. כמו כן, ישנם רעיונות (כגון [Floyd [1971) לפתוח מערכות עזר לכתיבה והוכחה במקביל של תוכניות: המתכנת יכתוב את תוכניתו על יד המסוף - ומערכת תפקח ותעזור לו, תצביע על טעויות מיד עם כתיבתן ותספק הוכחת נכונות לתוצאה הסופית.

לסיכום, לשיטות אימות נודעת כבר היום השפעה פסיכולוגית וחינוכית על צורת כתיבת תכוניות, ונראה לי סביר שבשנים הבאות תתבטא גם בתהליכים של כתיבת תוכניות ותיקונן.

מקורות

References

Boyer et.al. [1975] Boyer, Elspas, Levitt. SELECT - A Formal System for Testing and Debugging Programs by Symbolic Execution. Proc. Conf. on Reliable Software, April, 1975, Sigplan Notices, June 1975; pp. 234-245.

Deutsch [1973] Deutsch, L.P. An Interactive Program Verifier, Ph.D. thesis, Dept. of Computer Sci. U. of California, Berkely, June,1973.

Floyd [1967] Floyd,R. Assigning Meaning to Programs. Proc. Symp. in Applied Math. Vol. 19, J.T.Schwartz(ed.) Amer. Math. Soc., 1967, pp. 19-32.

Floyd [1971] Floyd, R. Towards Interactive Design of Correct Programs. Proc. IFIP Cong., Vol. 1, North-Holland, Amsterdam, 1971, pp. 7-10.

King [1974] King, J.C. Symbolic Execution and Program Testing, IBM Research Report, RC 5082,Yorktown-Heights, N.Y.

Manna [1974] Manna,Z. Mathematical Theory of Computation, McGraw-Hill, N.Y. 1974.

Waldinger and Levitt [1974] Waldinger, R. and Levitt, K. Reasoning about Programs. Artificial Intelligence, Vol. 5, 1974, pp. 235-316.

תודתי נתונה לד"ר מיכאל רודה עבור עזרתו בהכנת מאמר זה.

NONCONVEX QUADRATIC PROGRAMMING PROBLEMS
(WITH LINEAR CONSTRAINTS) BY THE
CAPACITY METHOD

by
A. BONEH & A. GOLAN

1. INTRODUCTION

Let x be a point in the real N-dimensional space and consider the constrained quadratic programming problem:

(1.1) $$VAL=\max_x F(x)=H^T x-\frac{1}{2}x^T Gx ,$$

subject to the linear constraints:

(1.2) $$Ax\leq B$$

and the sign constraints:

(1.3) $$x\geq 0$$

where H and B are arrays of length N and M respectively; A is an MxN matrix, G is a symmetric NxN matrix, F(x) is the objective function and VAL is the optimal value of the objective function.

It can be assumed, without loss of generality, that the point x=0 is feasible i.e.:

(2) $$B\geq 0$$

If x=0 is not a feasible point one can obtain (e.g by phase one of the simplex method) a point which is feasible (if no such point exists there is no solution to the given quadratic programming problem) and by appropriate transformation this feasible point can be shifted to the origin (for details see [2], page 378).

In a large number of applications the solution to this quadratic programming problem is required e.g. in mechanics [4], econometrica [1], system analysis [5].

A well known application with which this paper is concerned is the evaluation of the least square feasible point in the linear overdetermined set of equations:

(3) $Cx=D$

where C is a. LxN matrix, D is an array of length L, feasibility is defined by equations(1.2) and (1.3) and H and G are given by:

(4) $H=C^TD$

$G=C^TC$

The method considered in this paper can be used even in applications in which the variables are not constrained. In such case one artificial constraint should be introduced, i.e M=1, namely:

(5) $\sum_{i=1}^{N} a_i x_i \leq V$

where a_i (i=1,2...N) are nonnegative coefficients ($a_i \geq 0\ \forall i$) and V is a parameter known as the capacity parameter. From this parameter the method takes its name: The Capacity Method (CAPMET).

The sign constraints, $x \geq 0$, are essential in the use of CAPMET but there is no loss of generality if (all or some) of the variables can take negative values. In such case one can either:

1. apply to the unconstrained variables the transformation:

(6) $x_i' = x_i - s_i$

where s_i is a known lower bound on the optimal value of the i-th variable. Thus the transformed variables x_i' are sign-constrained as required for CAPMET i.e:

(7) $x_i' \geq 0 \quad \forall i$

or (if no prior information on s_i is known):

2. replace each unconstrained variable by the difference between two sign constrained variables, i.e:

(8) $x_i = y_i - z_i \qquad y_i \geq 0 \qquad z_i \geq 0$

this of course increases the dimensionality of the problem and with it, the computer time required to obtain the solution.

It is well known that the quadratic programming problem, eq. (1), can be classified according to whether the matrix G is:

1. Positive definite (or semidefinite)
2. Negative definite (or semidefinite)
3. Indefinite.

The capacity method is capable of solving the problem in any of these situations. However, global optimality is assumed only when G is positive definite (or semidefinite).

In the other cases the solution obtained may be a local optimum.

2. THE CAPACITY METHOD

2.1 Theoretical background

It is well known (e.g. see [6]) that the Kuhn-Tucker necessary conditions for optimality of the problem defined in eq. (1) are:

(9.1) $x \geq 0 \quad u \geq 0 \quad y \geq 0 \quad \lambda \geq 0$ sign constraints

(9.2) $x^T u + y^T \lambda = 0$ complementarity condition

(9.3) $Ax + y = B$

(9.4) $H - Gx = A^T \lambda - u$

where u is an N dimensional array (known as the dual variable) and y (known as the slack variable) and λ (the Lagrange multiplier) are arrays of length M.

The final row in eq. (9.3) is the capacity constraint and therefore the last element of the vector B is the capacity parameter v.

The solution of eq. (9) is a stationary point which can be local minimum, local maximum or a saddle point. In order to assure that the solution obtained is a local maximum certain second order conditions should also be fulfilled. It is worth noting that when G is positive definite (or semidefinite) these second order conditions are never violated. In all other cases the second order conditions might be violated and therefore it is important to see that these conditions will be fulfilled in the final solution [Ref. 6].

2.2 General description

The method was suggested by Houthakker [1] in 1960. The capacity method is an indirect parametric search (the capacity V being the parameter) and is not iterative in the usual sense. It is based on successive evaluation of a finite set of NK+1 critical capacity values V_i (i=0,1,2,...,NK) such that $0=V_0<V_1<V_2, \dots <V_{NK}=\infty$. In each step the

optimal point x* is evaluated as a function of v and the next critical capacity is obtained. (No starting point is required since for v=0 the only feasible point, and hence the optimal point, is x*=0). For all v in the i-th interval $v_{i-1}<v<v_i$ it is optimal to have:

(10) $$x_j^* = E_{ij} + v \cdot F_{ij} \qquad j=1,2,3,\ldots,N$$

where E_{ij} and F_{ij} are constants that change only when v increases through a critical value. The evaluation of E_{ij} and F_{ij} is carried out in a simplex-like tableau representing the Kuhn-Tucker necessary conditions for optimality. In additon to eq. (10), the capacity method enables to evaluate the objective function in the i-th interval which is given by $VAL=P_i+Q_i \cdot v+R_i v^2$ $i=1,2,3,\ldots,NK$ where P_i, Q_i and R_i are constants that change only when v increases through a critical value. The solution also contains the set of active constraints (if any) as a function of the capacity parameter v.

Clearly this method is very efficient in solving problems which are originally parametric in v i.e. problems in which the capacity constraint is built in the problem. In problems in which the capacity contraint is introduced artificially,the final solution is obtained for v=∞ in which case the capacity constraint is inactive (excluding the case of an unbounded objective function).

2.3 Algorithm description

The first Simplex-like tableau represents equations (9.3) and (9.4) and is given by:

w \ z	x	λ	u	y	y_c	VALUES OF BASIC VARIABLES E_0	F_0
u	-G	$-A^T$	I	0	0	-H	0
λ	A	0	0	I	0	B	0
λ_c	a	0	0	0	1	0	1

where y_c is the slack variable corresponding to the capacity constraint, λ_c is the Lagrange multiplier corresponding to the capacity

constraint, E_o and F_o are the independent term and the v-dependent term as defined in eq. (10) and a is the N dimentional array (1,1,1,...,).

The basic variables in each step of the algorithm are denoted by w_k (k=0,1,2,...,N+M) and the non-basic variables are denoted by z_k such that w_k and z_k are orthogonal variables i.e. either the pair x_k and u_k or the pair y_k and λ_k. (w_o will stand for the basic variable λ_c or y_c and z_o will stand for the non-basic variable λ_c or y_c). (w_k,z_k) will denote a pivoting operation in which the basic variable w_k is replaced by the non-basic variable z_k. P_{kj} will denote the j-th element in the k-th row of the tableau.

The Algorithm

Set i=0

Step 1: If i=0 define $a_k=1$ for k=1,2,...,N .

If i>0 define $a_k=\begin{cases}1 & \text{if } k\epsilon J\\ 0 & \text{else}\end{cases}$

Step 2: $w_r=\underset{k:w_k<o}{\text{MIN}}(w_k)$

and carry out the double pivot operation $(w_o,z_r),(w_r,z_o)$. as a result the tableau contains now the solution for v=0.

Set $J=\Phi$

Step 3: Set $i\to i+1$

if $F_{ij}\geq 0$ $\forall j$ go to step 5.

if there exists at least one j for which $F_{ij}<0$ define r by:

(11) $$v_i=\frac{E_{ir}}{F_{ir}}=\underset{\substack{j\notin J\\ F_{ij}<0}}{\text{MIN}}\{\frac{E_{ij}}{F_{ij}}\}$$

if $p_{rr}<o$ carry out the pivot operation (w_r,z_r) and repeat Step 3.

if $p_{rr}=o$ go to Step 4.

if $p_{rr}>0$ and w_r is either a u or a λ variable define $J=J+\{r\}$ and repeat Step 3.

if $p_{rr}>o$ and w_r is either an x or a y variable go to Step 1.

Step 4: define ℓ by:

(12) $$\frac{E_{i\ell}+F_{i\ell}v_i}{p_{\ell r}} = \underset{j:p_{jr}<0}{\text{MIN}} \left\{\frac{E_{ij}+F_{ij}v_i}{p_{jr}}\right\}$$

and carry out the double pivoting: $(w_\ell, z_r), (w_r, z_\ell)$
go to Step 3. If no j exists s.t. $p_{jr}<0$ go to step 1 .

Step 5: if $J \neq \Phi$ go to Step 1
if $J=\Phi$ then stop:
if $\lambda_c>0$ the objective function is unbounded.
if $y_c>0$ final solution is obtained.

3. NUMERICAL EXAMPLE

The capacity method was programmed in this study in Fortran IV and was tested on a CDC 6500 computer. Several examples of various sizes including positive definite, negative definite and indefinite problems were tested. In most cases the matrix G and the array H consisted of elements drawn at random from a uniform population.

As an example we illustrate one case with N=18 M=1 and in which the matrix G is indefinite with the following eigen values:

.0867 .0351 .0295 .0223 .0210 .0167 .0142 .0076 .0021 -.0068 -.0096 -.0154 -.0196 -.0286 -.0311 -.0329 -.0423 -.0491.

i.e. 9 eigen values are positive and 9 eigen values are negative. The constraints in this example were:

(13) $$\sum_{i=1}^{18} x_i \leq 1 \;;\; x_i \geq 0 \quad \forall i$$

This example requires two cycles through Step 1 i.e. one change ot the capacity constraint. The time required for a complete solution was 19.5sec.

Figure 1 represents the optimal value of the active variables as a function of the capacity parameter v. Variables not in the figure are non-basic and hence vanish for all v.

Figure 2 represents the objective as a function of v which is a non-decreasing function.

4. CONCLUDING REMARKS

The type of problem at hand depends solely on the matrix G. Therefore the eigen values of G are computed before the capacity method is initiated. If G is positive definite (or semidefinite) no change of the capacity constraint is required and hence the time required for the solution is minimal. Numerical experience with indefinite G matrices pointed out that only rarely a change of the capacity constraint occured. This feature is a great advantage of the algorithm since the computer time required in a typical indefinite quadratic programming problem is not much greater that required for a convex problem.

The dimensionality (or size) of the problem considered is given by N+M. Numerical experience with specific problems up to N+M=150 is promissing at least in the sense that the computer time required is of the order of 100-1000 secs. and hence is not prohibitive.

R E F E R E N C E S

[1] H.S. Houthakker: "The Capacity Method of Quadratic Programming" Econometrica, Vol. 28 (1960) pp. 62-87.

[2] C. Van De Panne & A. Whinston, "A Parametric Simplicial Formulation of Houthakker's Capacity Method, "Econometrica, Vol. 34 (1966) pp. 507-527.

[3] A. Majthay & A. Whinston & J. Coffman "Local Optimization for Nonconvex Quadratic Programming", Vol. 21, No.3, ~~Sept.~~ 1974, pp. 465-490.Naval Research Logistics Quarterly.

[4] G. Mayer. "A Quadratic Programming Approach for Certain Classes of Nonlinear Structure Problems" MECHANICA 3 (1968) pp. 121-130.

[5] M.H. Diskin & A. Boneh. "Properties of the Kernels for Time Invariant, Initially Relaxed, Second Order, Surface Runoff Systems" J. of Hydrology 17 (1972) pp. 115-141.

[6] A. Majthay,"Optimality Conditions for Quadratic Programming", Mathematical Programming 1, pp. 359-365 (1971).

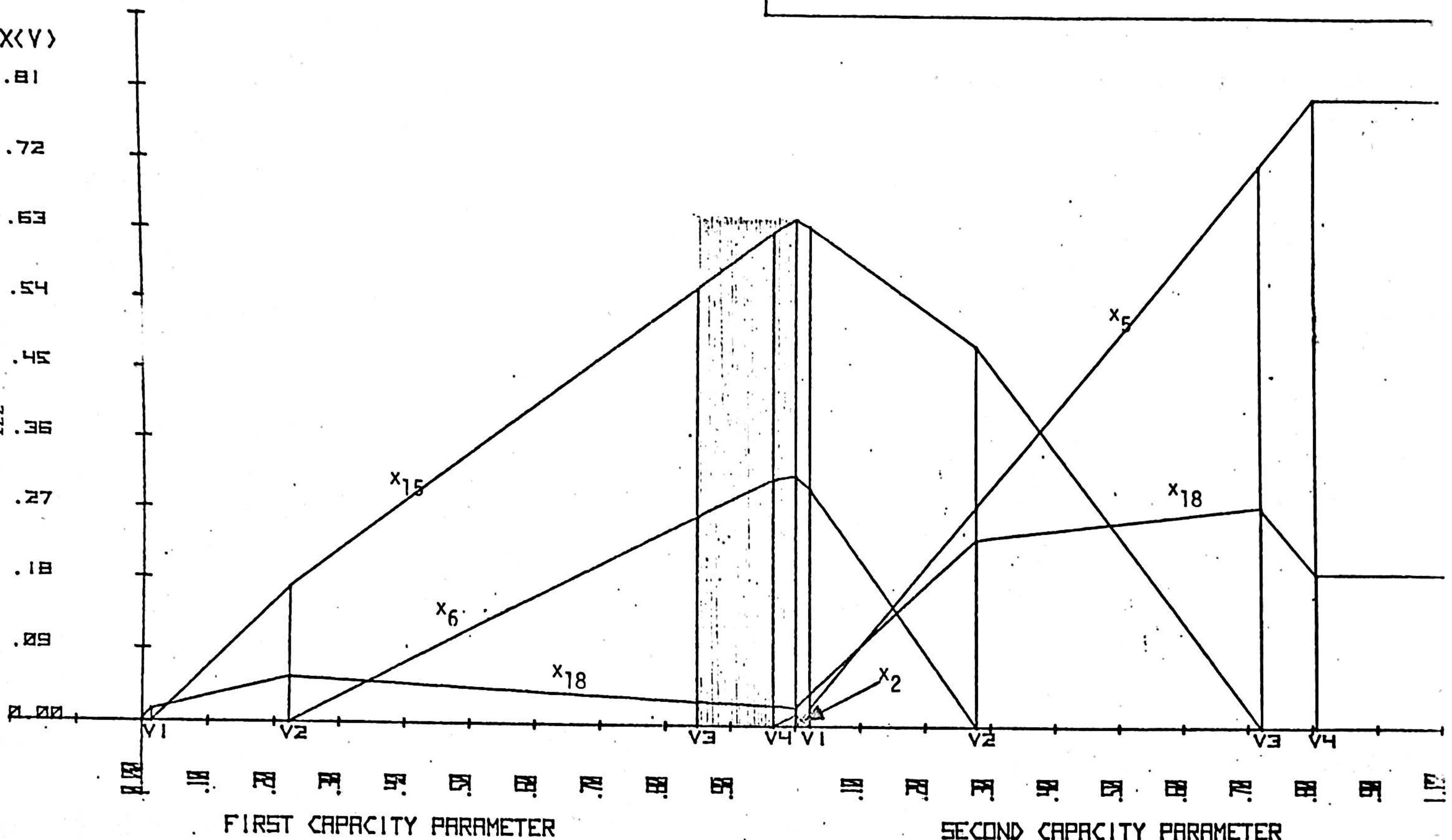


FIGURE 1: OPTIMAL VALUES OF THE ACTIVE VARIABLES AS FUNCTION OF THE CAPACITY PARAMETER V

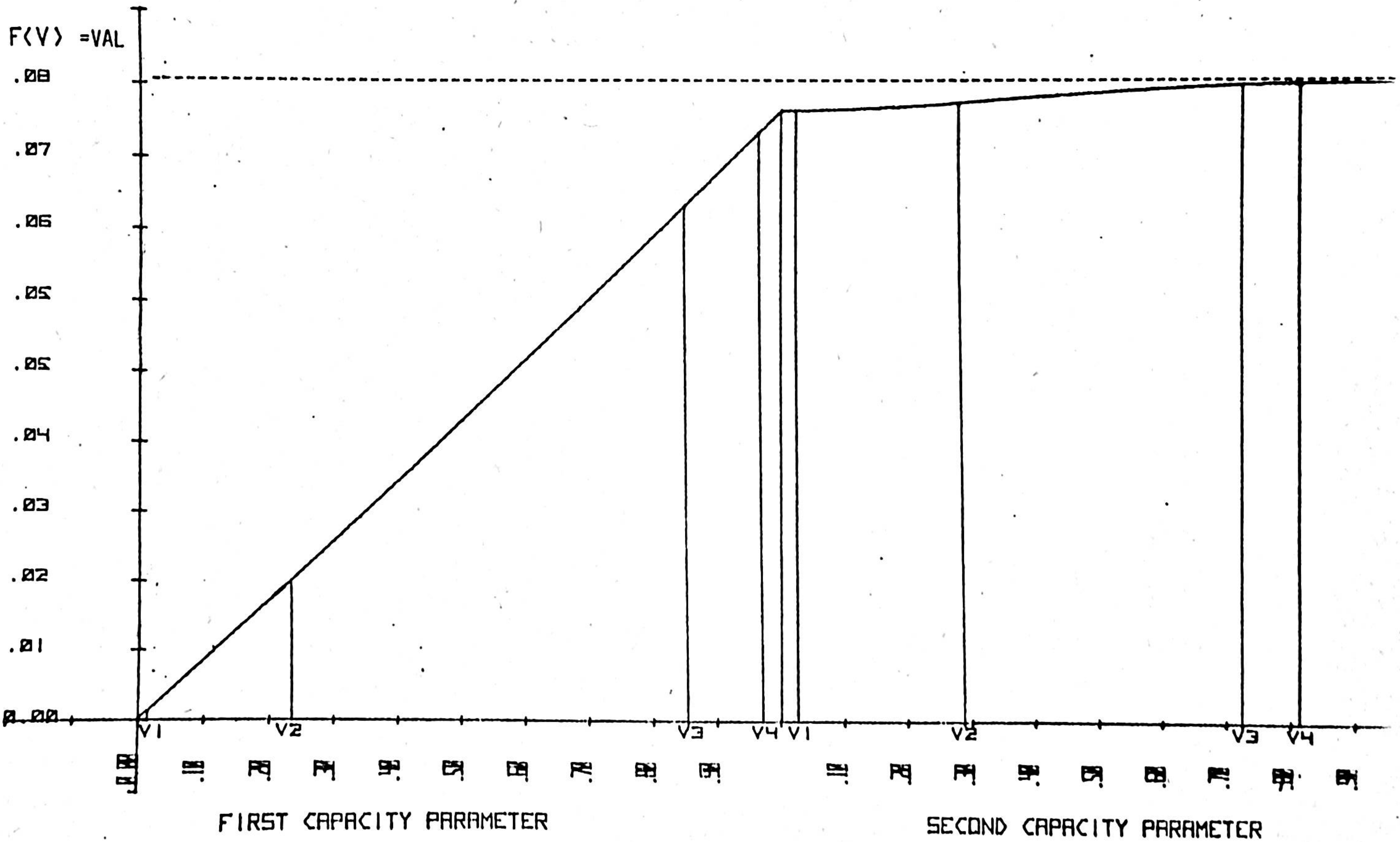


FIGURE 2 : OPTIMAL VALUE OF THE OBJECTIVE FUNCTION AS A FUNCTION OF THE CAPACITY PARAMETER V

A LINEAR ALGORITHM FOR FINDING REPETITIONS AND ITS APPLICATIONS IN DATA COMPRESSION

by

M. Rodeh* , V.R. Pratt** and S. Even***

ABSTRACT

A linear algorithm for finding the longest repetition for each position in a given string is developed. It is related to Weiner's string processing algorithm [1] but simpler. The algorithm is applied to the problem of evaluating the complexity of a finite string as defined by Lempel and Ziv [2]. Using Elias' representation of the integers [4,5] we produce an asymptotically optimal variable to variable coding scheme for sequential data compression based on the definition of complexity. The scheme requires unbounded memory for strings whose length grows to infinity. To overcome this difficulty Ziv and Lempel [3] invented a variable to block coding scheme that is asymptotically optimal for all ergodic sources with a given finite entropy. We implement their scheme by combining our repetition finder with Weiner's suffix tree construction algorithm. Other related and more practical schemes are described too.

* Dept. of Computer Science, Technion, Haifa, Israel.
Present address: IBM Israel Scientific Center, Technion City, Haifa, Israel.

** Dept. of Electrical Engineering and Computer Science, Mass. Ins. of Tech. Cambridge, Mass. 02139.

***Dept. of Computer Science, Technion, Haifa, Israel.
Present address: Computer Science Division - EECS University of California, Berkeley, California 94720.

0.1. Introduction.

This paper comprises 3 sections. In Section 1 we throw new light on Weiner's string processing algorithm [1] by developing a related algorithm for finding repetitions in a string. In Section 2 we adapt it to constitute a linear implementation of a variable-to-variable data compression scheme based on Lempel and Ziv's [2] definition of the complexity of a finite sequence; this scheme uses unbounded memory. A linear implementation of Ziv and Lempel's variable-to-block coding scheme with bounded memory is described in Section 3; it is a combination of our repetition finder and Weiner's algorithm [1] for suffix tree construction.

0.2. Notation.

Let Σ be a finite alphabet and Σ^* the set of all finite length strings over Σ. We use the letters a,b,c,... for elements of Σ and $\alpha,\beta,\gamma,\ldots$ for strings. We let ε denote the null string, $\ell(\alpha)$ the length of α, $\alpha\langle i\rangle$ the i-th letter of α, and $\alpha\beta$ the concatenation of α and β.

Let position in a string denote "between-letter" rather than "at-letter" position. Thus α has $\ell(\alpha)+1$ positions numbered $0,1,\ldots,\ell(\alpha)$. Two positions i and j of α define the subword $\alpha\langle i+1\rangle\ldots\alpha\langle j-1\rangle\alpha\langle j\rangle$ which occurs at j and is denoted by $\alpha\langle i,j\rangle$; if $i\geq j$ then $\alpha\langle i,j\rangle$ is defined to be ε. $\beta=\alpha\langle 0,i\rangle$ for $0\leq i\leq\ell(\alpha)$ is called a prefix of α; $\gamma=\alpha\langle i,\ell(\alpha)\rangle$ is called a suffix of α.

A repetition at j is an occurrence at j of a subword that also occurs at $i<j$; thus every occurrence but the first of a subword is a repetition.

An occurrence of $a\alpha$ at i is the left neighbor of the occurrence of α at i, and the occurrence of αb at i+1 is the right neighbor.

1. Improvements to Weiner's Algorithm.

1.1. Introduction.

In this section we develop an algorithm for finding repetitions within a string. It is related to Weiner's algorithm but is simpler. We also show that the running time of the algorithm on a RAM can be made independent of the alphabet size. To this end we establish properties of strings which prove the correctness of the algorithm.

1.2. Weiner's Suffix Identifier Finder.

Let α be a fixed string of length n. The suffix identifier of position i is the shortest (unique) subword which occurs only at i. To make this definition meaningful we have to add a new alphabet letter \$ to the left of α. Weiner's algorithm finds all suffix identifiers in O(n) time. (Weiner actually considered prefix identifiers, but the dual case is closer to our algorithm.)

We are interested in repetitions, which is a related concept. Out of the $\binom{n+1}{2}+1$ subwords of α, $O(n^2)$ different subwords may be repetitions. Thus in aiming at a linear

repetition-finding algorithm we cannot afford to identify every repetition explicitly. The. algorithm finds only the longest repetition at each position.

Observe that the longest repetition at position i depends only on $\alpha\langle 0,i\rangle$ and on nothing to the right of i. Thus in the producion of longest repetitions in a left-to-right scan, we never have to change the repetitions already found. This is a significant property since in our application we are interested in the scanning process itself and not in the final result. This is in contrast to the situation with the suffix identifiers which depend on the entire α. Thus in Weiner's algorithm when scanning α from left-to-right, some subword which occurs at j>i might cause an update operation which includes the suffix identifier for i, though this does not compromise its O(n) time bound.

The suffix identifier for i is longer than the longest repetition at i. However, scanning the string left-to-right, when we reach position i, they differ only by the leftmost letter of the suffix identifier which does not appear in the longest repetition.

It is worth noting here that neither Weiner's nor our algorithm proceeds from position i to position i+1 at constant time although both operate within a time bound linear in n.

1.3. An Algorithm for Finding Repetitions.

As pointed out above, we are interested in longest repetitions only. For example, in abcabcbc, the longest repetitions at each position 0 through 8 respectively are $\varepsilon,\varepsilon,\varepsilon,\varepsilon$,a,ab,abc,b,bc. We shall always use <u>longest repetition</u> to mean "longest repetition at a given position".

Notice that in this example later longest repetitions do not start in earlier positions. This is always true:

<u>Theorem 1</u>: Let β,γ be longest repetitions at i,j respectively. If i<j (i.e. if β ends before γ) then γ does not start before β.

<u>Proof</u>: Any prefix of this occurrence of γ must be a repetition. Thus if γ started before β, β could not be a longest repetition. □

This theorem suggests the basic algorithm, which simply moves two pointers (i.e. increments two counters) through α. We represent the pointers as brackets and call the in-between subword <u>bracketed</u>.

<u>Algorithm 1.1</u>: <u>start with</u> []α;
<u>while the picture is</u> ...[β]a... <u>do</u>
<u>if</u> βa <u>is not a repetition then move</u> [
<u>else move</u>].

"move [" and "move]" mean move the indicated bracket one position to the right. If ε is bracketed, "move [" also implies "move]" to avoid having "[" overtake "]".

Notice that when "]" first arrives at a position, the

bracketed subword is the longest repetition for that position.
One question remains: How do we tell efficiently whether βa is a repetition?

1.3. Testing for repetitions.

A remarkable property of longest repetitions supplies the key to testing whether βa is a repetition.

Lemma 1: If no occurrence of β in a string is a longest repetition then every occurrence of β has the same left neighbor.
Proof: We use induction on the number of occurrences of β in the string. The Lemma is vacuously true for zero or one occurrence. If there are two or more, the rightmost occurrence must have the same left neighbor as some other occurrence of β, since it is not a longest repetition. By induction all occurrences have the same left neighbor. □

Putting the Lemma in its contrapositive form we have:

Lemma 1': Let β occur at i and at j, $i<j$, not with the same left neighbor. Then there exists $k \le j$ such that β is the longest repetition at k.
Proof: Apply Lemma 1 to the first j characters of the string. □

Theorem 2: If βa is a longest repetition at j, there is some $k<j$ such that β is the longest repetition at k.
Proof: Let the left neighbor of the occurrence of βa at j be $b\beta a$. There must be an earlier occurrence of βa (since the one at j is a repetition) whose left neighbor is not $b\beta a$ (since the occurrence of βa at j is a longest repetition). Then β satisfies the conditions of Lemma 1', yielding the desired k. □

In order to test whether βa is a repetition (and therefore a longest repetition by the correctness of Algorithm 1.1), we ask whether β is a previously recorded longest repetition, and if so whether some earlier occurrence of β has right neighbor βa. To answer the first question we keep a record of longest repetitions seen so far, represented as vertices of a graph G. Each vertex corresponds to a distinct element β of Σ^*, and so represents all repetitions of a given subword β. In our discussion we refer to a vertex by underlining the string which it represents. To answer the second question, we associate with each vertex $\underline{\beta}$ a record for each letter a of whether βa has occurred, and if so where it first occurred. We let $\underline{\beta}.a$ name a register associated with $\underline{\beta}$, which we may inspect and update. If $\underline{\beta}.a$ is undefined, βa has not occurred before; if defined, βa first occurs at $\underline{\beta}.a$. (The single dot suggests the first occurrence.) There are $|\Sigma|$ such registers associated with each vertex, all initially undefined.

We postpone for the moment the question of how we keep this information about right neighbors up to date, as it depends on the details of how we navigate in and update the graph G.

In order to locate β when asking whether βa has occurred before, we shall maintain a pointer to the vertex in G representing the currently bracketed subword. Clearly we need enough information in G to allow us to trace a trajectory through G corresponding to the sequence of bracketed words resulting from moving the brackets.

By Theorem 2, Algorithm 1.1 must keep moving "[" until some word is bracketed for which there is a vertex in G. Hence for each vertex $\underline{\beta}\neq\underline{\varepsilon}$ in G, all we need is an edge (which we label S for suffix) from $\underline{\beta}$ to the vertex which represents the longest proper suffix of β in G. We denote by S($\underline{\beta}$) a register associated with $\underline{\beta}$ containing this edge (i.e. pointer). In this way, we may jump directly from $\underline{\beta}$ to S($\underline{\beta}$), following possibly several moves of "[".

When we move "]", we go from $\underline{\beta}$ to $\underline{\beta a}$. If $\underline{\beta a}$ is already in G, an edge (labelled :a) from $\underline{\beta}$ to $\underline{\beta a}$ suffices; for each letter a we associate with $\underline{\beta}$ a register named $\underline{\beta}$:a to contain $\underline{\beta a}$. (The two dots suggests that βa has occurred at least twice). If $\underline{\beta a}$ is not yet in G, we must create it. This is sufficiently complicated to warrant its own section.

Except for the details of vertex creation, and of keeping $\underline{\beta}$.a up to date for all a and for all $\underline{\beta}$, we may now flesh out Algorithm 1.1 to yield Algorithm 1.2.

In the following, β denotes the currently bracketed subword and i its position (thus i represents "]"). The function GENERATE returns a new vertex, with all associated registers undefined. Note that we do not have to represent "[" explicitly β keeps track of it implicitly.

<u>Algorithm 1.2</u>:

```
Initialize      ε←GENERATE;
                β←ε; i←1;
Iterate         while i≤n do
                      begin a←α<i>;
Repetition?           if β.a undefined then
Move [                   if β≠ε then β←S(β)
  or []                          else i←i+1
Create?           else begin if β:a undefined then CREATEVERTEX;
                            β←β:a; i←i+1
                            end
                      end;
```

We turn now to vertex creation, and postpone the details of updating right neighbors for the moment.

1.5. Creating Vertices.

Merely generating a new vertex $\underline{\beta a}$ is easy, as is updating $\underline{\beta}$:a. It is more difficult to correctly update the S edges leading to and from $\underline{\beta a}$. By definition only one edge leads from $\underline{\beta a}$. Fortunately, when it is created, at most one edge leads to $\underline{\beta a}$, as Corollary 3.2 shows.

<u>Theorem 3</u>: Given a and τ, there exists at most one subword π such that $\underline{\pi a \tau}$ is in G and S($\underline{\pi a \tau}$)=$\underline{\tau}$.

Proof: Let $\pi a\tau$ and $\pi' a\tau$ be two such distinct subwords, both represented in G. Neither may be a suffix of the other, since $S(\underline{\pi a\tau})=S(\underline{\pi' a\tau})=\underline{\tau}$. Hence $\pi a\tau$ and $\pi' a\tau$ must be of the forms $\eta f\rho a\tau$ and $\eta' g\rho a\tau$ ($f\neq g$) respectively where any of η, η' and ρ may be null. Thus by Lemma 1, $\rho a\tau$ occurs as a longest repetition no later than the later occurrence of $\pi a\tau$ and $\pi' a\tau$, contradicting $S(\underline{\eta f\rho a\tau})=\underline{\tau}$. ∎

Corollary 3.1: For each letter c and vertex $\underline{\beta}$ we may introduce an edge (register) labelled $*c$: from $\underline{\beta}$ to the (necessarily unique) vertex $\underline{\gamma}$ of the form $\psi c\beta$ such that $S(\underline{\gamma})=\underline{\beta}$ (The "$*$" denotes the ψ in $\psi c\beta$). Thus $S(*c:\underline{\beta})=\underline{\beta}$, provided $*c:\underline{\beta}$ is defined.

Corollary 3.2: When creating a vertex $\underline{\beta a}$, at most one vertex $\underline{\gamma}$ exists such that $S(\underline{\gamma})=\underline{\beta a}$.
Proof: By Lemma 1 all previous occurrences of βa must have the same left neighbor $b\beta a$, if any. Hence, if βa were a suffix of a bracketed subword, it should have $b\beta a$ as left neighbor and by Theorem 3, there exists at most one vertex $\underline{\gamma}$ such that $S(\underline{\gamma})=\underline{\beta a}$.

Hence our problem is reduced to that of finding the $\underline{\gamma}$ which is $S(\underline{\beta a})$, and the $\underline{\delta}$, if any, for which $S(\underline{\delta})=\underline{\beta a}$, and updating all the edges between $\underline{\delta}$, $\underline{\beta a}$ and $\underline{\gamma}$.

Either $\gamma=\varepsilon$ or $\gamma=\gamma' a$, for some γ'. If the latter, then by Theorem 2, since $\underline{\gamma}$ is in G, so is $\underline{\gamma'}$. To find $\underline{\gamma'}$ we simply follow along S edges, starting from $\underline{\beta}$, until we find a vertex $\underline{\pi}$ for which $\underline{\pi}:a$ is defined; $\underline{\pi}:a$ is the required $\underline{\gamma}$. If there is no such $\underline{\pi}$ (the searce ends with $\underline{\pi}=\underline{\varepsilon}$ and $\underline{\varepsilon}:a$ is undefined) then we are in the first case and $\underline{\gamma}=\underline{\varepsilon}$.

To find $\underline{\delta}$, note that before $\underline{\beta a}$ is created, $S(\underline{\delta})=\underline{\gamma}$. If $\beta a=\eta c\gamma$ for some η and c, then $\underline{\delta}=*c:\underline{\gamma}$. To see that $*c:\underline{\gamma}$ must yield a $\underline{\delta}$ of which βa is a suffix, note that if it didn't, then $*c:\underline{\gamma}$ would have to point to both $\underline{\beta a}$ and $\underline{\delta}$, contrary to Corollary 3.1.

So we wish to compute c. In the absence of a representation of c in G, we look in α. For this purpose we introduce two registers , for each vertex $\underline{\beta}$; $loc(\underline{\beta})$ gives the location of the first occurrence of β and $len(\underline{\beta})$ holds the length of β.

Now we can easily find c. Let $j=loc(\underline{\beta a})-len(\underline{\gamma})$; then $c=\alpha\langle j\rangle$. When setting $*d:\underline{\beta a}$ to $\underline{\delta}$ we find the d in the same way except that $j=loc(\underline{\delta})-len(\underline{\beta a})$, $d=\alpha\langle j\rangle$.

The remaining question is that of updating right neighbors of subwords represented in G.

1.6. Updating Right Neighbors.

For the algorithm's correctness it is enough to show a way to update $\underline{\beta}$'s right neighbors no later then the first occurrence of $[\beta]a$. We keep β's right neighbors up to date from the moment of creating $\underline{\beta}$. This is useful in certain applications.

First assume that the earliest occurrence of βa is later than the earliest bracketed occurrence of β. Then when] first

arrives between β and a, [must be to the left of this occurrence of β. Now] may not move while [is to the left of β since this is the first occurrence of βa. But in the process of moving to the right, [must come to bracket β since $\underline{\beta}$ is already in G. At this time we are in a position to define $\underline{\beta}$.a, which can be done with $\underline{\beta}$.a←i. By only performing this assignment when $\underline{\beta}$.a is undefined, we avoid erroneously redefining $\underline{\beta}$.a at a non-initial occurrence of βa.

Now assume that the first occurrence of βa is not later than the first bracketed occurrence of β. We have to show that when $\underline{\beta}$ is created we can immediately record the position of the first βa.

Two cases arise:

(i) The second occurrence of β is bracketed. In this case, only one right neighbor is involved, namely that of the first occurrence of β. Thus when β is first bracketed, loc($\underline{\beta}$) may be used to determine this right neighbor. In this case there exists no δ such that S($\underline{\delta}$)=$\underline{\beta}$, since the presence of such a δ implies at least two earlier occurrences of β.

(ii) The third or later occurrence of β is the first to be bracketed. In this case, there must be some δ such that S($\underline{\delta}$)=$\underline{\beta}$, because the second occurrence of β can avoid being bracketed only by not being the longest repetition at that position. In this case, there may be several right neighbors to be recorded; however, every previous occurrence of β must be at the same position as some occurrence of δ, and since all of δ's right neighbors are up to date (by induction), we can simply copy their positions, which will then serve as the positions of the corresponding right neighbors of β.

Now we collect all the pieces to yield the following final algorithm.

1.7. The Algorithm in Full Detail.

Algorithm 1.3

```
Initialize    ε←GENERATE;
              len(ε)←0;
              loc(ε)←0;
              S(ε)←ε;
              β←ε;
              i←1;
Iterate       while i≤n do
              begin
              a←α<i>;
Repetition?   if β.a undefined then
New rt. nbr      begin β.a←i;
Move [                 if β≠ε then β←S(β)
  or []                        else i←i+1
                 end
Create?       else begin if β:a undefined then CREATEVERTEX;
Move ]                  β←β:a;
                        i←i+1
                    end;
              end;
```

```
CREATEVERTEX: βa←GENERATE;
              len(βa)←len(β)+1;
              loc(βa)←β.a;
Link β,βa     β:a←βa;
Find γ        γ'←S(β);        comment: we need S(ε)=ε here;
              while γ':a undefined and γ'≠ε do γ'←S(γ');
              γ←if γ':a defined and β≠ε then γ':a else ε;
Find δ        j←loc(βa)-len(γ);
              c←α<j>;
              δ←*c:γ;
Link βa,γ     S(βa)←γ;
              *c:γ←βa;
              if δ defined then begin
Link δ,βa                         j←loc(δ)-len(βa);
                                  d←α<j>;
                                  S(δ)←βa;
                                  *d:βa←δ
                                end;
Rt. nbrs      if δ undefined then βa.α<β.a+1>←β.a+1
                              else for each letter b do
                                   βa.b←δ.b; end
```

1.8. Timing.

We wish to show that the running time of the algorithm is linear in $n=\ell(\alpha)$. We assume that the alphabet is fixed and finite.

The main routine moves one or both brackets to the right at each iteration of the single loop. Hence the body of that loop may not be executed more than 2n times.

CREATEVERTEX contains two loops. The second, which transcribes right neighbors is executed $|\Sigma|$ times, thus constant. The first loop searches for γ'. The search is done by moving from β along the S arcs. We introduce a third pointer, in addition to the two brackets; it points to the starting position of the occurrence of γ' at i, during this search. We claim that this pointer never moves left through α, even between successive vertex creations, and moves right at every iteration of the search loop. The latter claim is non-controversial. To see the former, observe that when the search that started from β terminates, a search that starts from βa must begin with the just-found γ. A similar observation applies even when the next search does not start from βa. This establishes the linear time bound for a finite alpabet. In fact the algorithm is linear independently of the alphabet size, but we shall not prove it here.

2. A Variable to Variable Coding Scheme for Sequential Data Compression.

2.1. Introduction.

In this section we present a linear scheme for sequential data compression in two steps:

Step 1: We design an algorithm for translating strings to words

over a certain infinite alphabet Θ. The translation method is based on Lempel and Ziv's definition of the complexity of finite strings [2]. The implementation uses the repetition finding algorithm. The handling of the data structures is irrelevant here. Therefore we shall refer to Algorithm 1.1 and not to the full detailed version. The linearity will come out as an immediate consequence of the timing considerations of the repetition finding algorithm.

<u>Step 2</u>: We use Elias' universal binary representation of the integers [4,5] to produce a binary encoding of strings over Θ. For a given integer m this representation can be built within a time bound linear in the length of the binary representation of m.

The composition of the two steps yields a universal linear variable to variable encoding scheme for strings, whose compression ratio tends to be optimal, for ergodic sources, as the length of the input string grows to infinity. We shall prove this by referring to results of Lempel and Ziv [3]. The disadvantages of the scheme are that the memory requirements grow with the length of the input string and the rate of convergence to the optimal compression ratio is slow. We shall treat the memory problem in Section 3.

2.2. Translating strings to Θ^*.

Let $\Theta = \{\langle i,j,a\rangle \mid i,j \text{ are integers}, a \in \Sigma\}$. The letters of Θ are triples whose first two components are integers, and the third is a letter of Σ. A word over Θ is a sequence of such triples.

Define the translation of α to Θ^*, denoted by $T(\alpha)$, by the following algorithm:

<u>Algorithm 2.1.</u>

```
p←0; k←0;
while p<ℓ(α) do
begin m←maximum integer for which there exists a j<p
      satisfying α<j,j+m>=α<p,p+m> and p+m<ℓ(α);
      i←some j satisfying the above;
      k←k+1; p←p+m+1;
      t_k←<i,m,α<p>>
end;
```

It can easily be seen that $T(\alpha)$ exists and is well defined. Lempel and Ziv [2] define $\ell(T(\alpha))$, the number of triples in the translation of α, which is the same as the final value of k in Algorithm 2.1, to be <u>the complexity of</u> α; denote it by $C(\alpha)$.

Assume that we have translated $\alpha\langle 0,p\rangle$ into $t_1 \ldots t_k$, and computed $t_{k+1} = \langle i,m,\alpha\langle p+m+1\rangle\rangle$. Thus $\alpha\langle p,p+m\rangle$ is a repetition at $p+m$. If $p+m<\ell(\alpha)$ then $\alpha\langle p,p+m+1\rangle$ is not a repetition. In terms of the brackets movements of Algorithm 1.1 we may say that when "]" first arrives at p+m "[" is not to the right of p while when "]" is at p+m+1 "[" is to the right of p. The following algorithm implements the ideas above:

We let $p_{[}, p_{]}$ denote the current position of [and] respectively; loc(β) the location of the first occurrence of β, and p the position immediately *after* the current translated prefix of α.

Algorithm 2.2.

```
k←0; p←0;
start with []α;
while the picture is ...[β]a... do
begin if βa is a repetition then move ] else
        if β≠ε then begin j←p]-p;
                          i←loc(β)-j;
                          move [;
                          if p[>p then begin
                                             k←k+1;
                                             tk←<i,j,a>;
                                             p←p]+1  end  end
              else begin if p]=p then begin k←k+1;
                                                tk←<0,0,a>;
                                                p←p+1  end;
                    move [ end  end;
if p<ℓ(α) then  begin j←ℓ(α)-p-1;
                      i←loc(β)-j;
                      k←k+1;
                      tk←<i,j,α<ℓ(α)>>  end;
```

It is easy to see that the movements of "[" and "]" on α are identical to those in Algorithm 1.1. The production of T(α) is added to the identification of longest repetitions and does not change the flow of the repetition finding algorithm. To each iteration of the single loop, we add a constant number of operations. Therefore, Algorithm 2.2 is as linear as Algorithm 1.1.

2.3. Encoding of Words Over Θ.

For simplicity we assume that $\Sigma=\{0,1\}$. Let $\tau=t_1 t_2 \ldots t_k$ be a word over Θ. Each t_q has the form <i,j,a>. We use Elias' representation of the integers to encode i and j. The code of t_q is formed by the concatenation of these codes and a. We get a uniquely decipherable binary encoding of τ.

Let B(n) denote that binary representation of n, for which the most significant bit is 1. LGx is an abbreviation for $\lceil \log(x+1) \rceil$ where for the binary case, the base of the logarithm is 2, and $\lceil x \rceil$ is the least integer $\geq x$; $LG^2 x = LG(LGx)$, $LG^{i+1} x = LG^i(LGx)$.

Define R'(n) as follows:

$$R'(n)=\begin{cases} b_1 b_2 b_3 & n \leq 7 \\ R'(LGn)B(n) & n>7 \end{cases}$$

where $b_1 b_2 b_3$ is a binary representation of n.

Next we define R(n) by R(n)=R'(n)0. R(n) is the representation of n which we use, and it has the prefix property (there exists no m n for which R(m) is a prefix of R(n)). It can be proven [5] that for large n,

$$\ell(R(n))=1+\sum_{i=1}^{LG^i n=3} LG^i n$$

and $\ell(R(n))\leq LGn+2LG^2 n.$

The cost of producing $R(n)$ is $O(\ell(R(n)))$. But $\ell(R(n))\leq 2LGn$. Therefore, the production time of $R(n)$ is linear in the length of the binary representation of n.

2.4. Encoding of Strings.

We can show that the compression ratio of the composed encoding scheme tends to be optimal as the length of α grows to infinity but we omit the details.

Clearly, the main handicap of this data compression scheme is that the memory requirement is unbounded. In the next section we shall describe a method more suitable in cases of bounded memory.

3. Algorithm for Data Compression with Bounded Memory.

3.1. Introduction.

Ziv and Lempel [3] invented a compression method which first parses an input string into bounded length substrings and then uses an adaptive coding scheme which maps these substrings sequentially into uniquely decipherable code-words of fixed length over the desired alphabet. For a given ε and for all finite memory sources with entropy H, the scheme uses a fixed amount of memory and its compression ratio is proven to be close to optimal up to ε.

We implement the proposed compression method by combining Weiner's algorithm with the repetition finding algorithm described in Section 1. The two parameters of the algorithm are N, the memory size, and F, a bound on the length of source substring to be represented by a single code word.

3.2. The Compression Method.

Let α be an input string. We shift a window of size N along α. It is set initially to hold zeros and a prefix of length F of α. We encode α step by step. At each step we assume that the N-F leftmost letters within the window have been encoded already and we encode as much as we can out of the rightmost F letters. Let α_i denote the longest prefix of these F letters such that there exists an earlier occurrence of $\alpha_i\langle 0,\ell(\alpha_i)-1\rangle$ within the window. Denote by p_i the start position of such an occurrence (the window's left boundary is considered to be position 0). Then we encode α_i into

$$C_i=\langle p_i,\ell(\alpha_i)-1,\alpha_i\langle\ell(\alpha_i)\rangle\rangle,$$

and shift the window $\ell(\alpha_i)$ positions to the right.

For simplicity assume that the source and code alphabets are binary. To write each C_i we use

$$L=\lceil\log_2(N-F)\rceil+\lceil\log_2(F)\rceil+1$$

bits (padding with zeros if necessary). The code of α is the concatenation of its C_i's.

3.3. Implementation of the Compression Method.

The compression method just described is similar to that in Section 2. There are two main differences:

(i) The length of each α_i is bounded by F. To handle this constraint all we have to do is to follow the "]" movements and when it goes "too far", produce a new C_i.

(ii) The size of the window is fixed. Thus when we shift its right end, we have to shift its left end too.

Algorithm 1.3 is designed to handle strings given in a left to right mode. It currently holds information about a portion of the input string which starts at a fixed leftmost position, P_1, and ends in a position that advances to the right. If the fixed start position lies inside the window, it can simulate the window's right side movements. In fact the suffix of length $F-\ell(\alpha_i)$ of the window's content is ignored in such a simulation.

Weiner's algorithm handles strings in a right to left mode. Assume that Weiner's algorithm is applied from some fixed rightmost position, P_2, to the left and proccessed K letters. While the algorithm is proceeding, a stack is built which holds all actions on the data structure. Using this stack we can "reverse the algorithm" and get a new algorithm which acts upon the string from left to right in linear time while its right end is fixed. (Actually only a substructure of the data is necessary in our application and the algorithm can be improved accordingly.) If this fixed position lies inside the window the reversed algorithm can simulate the window's left side movements.

To implement the above compression method we have to be able to check for the existence of an occurrence of a string whose length is bounded by F-1 within the window. Therefore we take $P_2=P_1+F-2$. In this way if there exists such an occurrence, it occurs either on the left side of P_2 or on the right side of P_1; thus it is treated by at least one of the two algorithms.

To ensure that there always exist two positions P_1 and P_2 inside the window we design a sequence of such couples: Note that the N-F leftmost letters are initially zero. To make the discussion uniform define ξ to be $0^{N-F}\alpha$ and put the window on the first N letters of ξ. In the sequel "position j" abbreviates "position j in ξ". Define P_1^i and P_2^i by:

$$P_1^1=\lfloor (N-F)/2\rfloor-(F-2)$$
$$P_2^1=\lfloor (N-F)/2\rfloor;$$

For $i\geq 1$

$$P_1^{i+1}=P_1^i+\lfloor (N-F)/2\rfloor$$
$$P_2^{i+1}=P_1^{i+1}-F+2.$$

Usually F<<N. More explicitly we assume that N>9F-14. In this case the number of couples inside the window is at most three; the leftmost is being used and the others are being developed. When the couple in use is shifted out, the one next to it is ready, and we start using it. At this moment, P_2^{i+1} is located at position $\lfloor (N-F)/2\rfloor$. Thus Weiner's algorithm has had enough time to produce the desired data structure. Therefore the composed algorithm is linear.

The assumption that N>9F-14 is sound since for a given

N, F should satisfy

$F \sim \log N/h$,

and, as N increases the ratio between F and N decreases logarithmically. Thus, a reasonable choice of N will satisfy the inequality.

REFERENCES

[1] P.Weiner - "Linear Pattern Matching Algorithms", IEEE 14th Annual Symposium on Switching and Automata Theory, pp.1-11, 1973.

[2] A.Lempel and J.Ziv - "On the Complexity of Finite Sequences", IEEE Trans. on Information Theory Vol. IT-22, pp.75-81, January 1976.

[3] J.Ziv and A.Lempel - "A Universal Algorithm for Sequential Data Compression", Faculty of Electrical Engineering, Technion, Haifa, Israel, 1975.

[4] P.Elias - "Universal Codewords Sets and Representations of the Integers", IEEE trans. on Information Theory Vol. IT-21, pp.194-203, March 1975.

[5] S.Even and M.Rodeh - "Economical Encoding of Commas between Strings", TR-54, Dept.of Comp. Scie., Technion, Haifa, Israel, July 1975.

[6] I.Shperling - "On the Asymptotic Complexity of Sequences", Faculty of Electrical Engineering, Technion, Haifa, Israel, December 1975.

[7] A.V.Aho, J.E.Hopcroft and J.D.Ullman - "The Design and Analysis of Computer Algorithms", Addison-Wesley, 1974.

קבוצת דיון מס' 10:

"עיבוד רחך - שיתוף זמן"

יו"ר: אל"מ מ. נדיר

צ. טל

ג. אריאב

א. כרמל

שתף זמן - ושימושים בו במחלקת מיכון

צבי של - בנק ישראל

מבוא

"מר סמית המשיך בצעידתו, ונכנס לאולם הכתבים. כ-1500 כתבים היו ממוקמים מול מספר זהה של מכשירי טלפון, מספרים למנוייהם את חדשות העולם, כפי שנאספו במשך הלילה.

הארגון של שירות ייחודי זה, תואר לעיתים קרובות. ליד מכשיר הטלפון של כל כתב, כפי שהקורא נוכח, יש סדרה של קומוטטורים (מחלפים) המאפשרים לו להתקשר לכל אחד מהקוים "הטלפוטים", באופן שהמנוי יכול לא רק לשמוע את החדשות אלא גם לראותן.

כאשר מתארים ארוע, תמונות מאפיינות מועברות יחד עם הכתבה, ואין כל השפעה של הנשמע על הנראה.

פרטי הדווח של הכתב, יחד עם יתרת התאור וקטעי העיתונות, מאופיינים בצורה אוטומטית, באמצעות המערכת הגאונית, ומגיעים למאזינים בעיתוי המתאים.

יתרה מזו, המאזינים חופשיים לבחור את הנושא המעניין אותם, שעה שהכתבים רשאים להתחבר או לסרב להתחבר לעורכיהם".

לא יאומן כי יסופר, אך חזון שתף-זמנים זה מופיע ב-THE FORUM של שנת 1889 תחת הכותרת "IN THE YEAR 2889". והמחבר אינו אחר מאשר - ג'ול וורן.

והנה אנו לא בשנת 2889, ומערכות מורכבות הרבה יותר מהמתוארת מופעלות בהצלחה.

היסטוריה

מקובל ליחס את תחילת התופעה עצמה שתף-זמן לכנס UNESCO שנערך בשנת 1959. בכנס זה הוגדרו מספר מאפיינים למערכת מחשב "שיתופית":

א. מתן פתרון לניצול יעיל של משאבי מחשב בתקופה בה גדלה והולכת מהירות העיבוד;

ב. ויסות רציף של בעיות (תכניות) לפתרון במחשב, בהנחה שיהא איבוד של מספר שניות בין בעיה אחת לשניה;

ג. ביצוע שתף-זמן בין מפעילים (משתמשים) באופן שבזמן שאחד המפעילים אינו מנצל את המחשב, ישתמש בו האחר;

ד. לכל אחד מהמפעילים יהא מחשב ל"כאורה", אם כי איטי מאשר אילו הפעילו באופן בלעדי.

גם בתקופה בה נוסחו מאפיינים אלו, הם נראו "נועזים".

2/.

צעד חשוב בכיוון פיתוח מערכות שתף-זמן נעשה במכון הטכנולוגי במסצ'וסטס (M.I.T) במסגרת פרויקט ה-CTSS - (COMPATIBLE TIME-STARING SYTEM). המערכת הוקמה על מחשב י.ב.מ. מטיפוס 709, והיתה מערך נסיוני אשר שופר מאוחר יותר על מחשב מטיפוס י.ב.מ. 7094 ונקרא CTSS II. הפרויקט שהונחה ע"י E.J. CORBATO אפשר גישה משותפת למחשב של 30 מסופים במקביל.

במקביל, ומעט מאוחר יותר פותחו מספר מערכות נוספות במרכזי מחקר ואוניברסיטאות:

א. BOLT, BERANEK, NEWMAN על מחשב PDP-1;

ב. RAND CORPORATION פתחה מערך בשם JOSS;

ג. DEVELOPMENT CORPORATION פתחה מערך בשם Q-32;

ד. והמפורסם באותה עת - DARTMOUTH COLLEGE פתח מערך שתף זמן על מחשב GE225.

שירותים מסחריים לשתף זמן החלו גם הם בתחילת שנות ה-60 ב-3 מקומות:

א. ADAMS ASSOCIATES ב-קיימברידז' ארה"ב על PDP-4;

ב. KEY DATA הקשורה לחברה הקודמת על UNIVAC 491;

ג. I.B.M הכריזה כבר ב-1964 על מערכת QUICKRAN על מחשבי 7040/7044.

כיום מפורסם פרויקט ה-MULTICS ב-MIT (MULTIPLEXED INFORMATION ON COMPUTING SERVICE) אשר הנו המשך ישיר של פרויקט ה-CTSS במסגרת ה-PROJECT MAC.

לפרויקט ה-MULTICS 4 מאפיינים אשר מרחיבים את משמעות שתף-הזמן:

א. דגש על תגובה מהירה, למספר רב של משתמשים;

ב. שימוש בגוון רחב של תוכנה בזמן אמיתי, כמו תכניות עריכה, למשל, שירותי ניפוי (DEBUG), קומפילרים שונים, ספריות ושגרות;

ג. התיחסות זהה לתכניות וקבצים, כאל מסת נתונים משותפת ברת גישה, למספר רב של משתמשים;

ד. אפשרות הוספת מאפיינים נוספים.

3/.

כאשר מדובר כיום על שתף-זמן, הכוונה בעיקר לחברות אשר מוכרות שירותי שתף-זמן.
כיום פועלות בארה"ב למעלה מ-200 חברות המוכרות שירותים כאלה ברשתות מקומיות,
ארציות ובין-ארציות. סקר שנערך ע"י DATAPRO פורסם במאי 1976 - ובו מספרים
מאלפים בנושא שתף-זמן:

א. טבלת התפלגות השימוש בשתף-זמן לפי נושאים:

1.	FINANCIAL	35%
2.	STATISTICAL	17%
3.	ENGINEERING	16%
4.	MODELING	14%
5.	DATA BASE MANAGEMENT	13%
6.	ACCOUNTS PAYABLE	10%
7.	ACCOUNTS RECIEVABLE	10%
8.	INVENTORY CONTROL	7%
9.	SCIENTIFIC	6%
10.	SIMULATION	6%
11.	PRODUCTION	4%
12.	PAYROLL	4%
13.	INSURANCE	4%
14.	GENERAL LEDGER	4%
15.	BANKING	3%
16.	INFORMATION RETRIEVAL	3%

ב. מבחינת אפיון המשתמשים:

MANAGEMENT	15%
PROGRAMMERS	25%
OTHER DATA PROCESSING PERSONEL	22%
ALL OTHERS	38%

ג. אופי השימוש בשירותי תקשורת למחשב:

100% CONVERSATIONAL (T/S)	51%
OVER 50% CONVERSATIONAL (T/S)	31%
OVER 50% REMOTE BATCH	13%
100% REMOTE BATCH	5%

4/.

ד. תוצאות סעניינות התקבלו גם ממחן "ציונים" לחכונות נבחרות. בשימוש במערכי שתף-זמן:

		EXCELLENT	GOOD	FAIR	POOR (%)
1.	OVERAL SATISFACTION	26	54	17	3
2.	COST EFFECTIVESS	18	41	29	12
3.	TECHNICAL SUPPORT QUALITY	31	45	20	4
4.	TECHNICAL SUPPORT AVAILABILITY	26	41	25	7
5.	SALES PERSONNEL QUALITY	26	48	19	7
6.	TRAINING EFFECTIVENESS	14	50	27	9
7.	EEAS OF USE FOR EXPERIENCED D.P. PEOPLE	45	46	8	1
8.	EEAS OF USE FOR INEXPERIENCED OR NON D.P. PEOPLE	18	46	28	8

נתרכז באחד ממרכיבי השימוש במערכות שתף-זמן, והכוונה לשימושים במח' מיכון.

<u>שתף-זמן בתכנות - הרחבה</u>

היעד העיקרי הניתן להשגה באמצעות מערך שתף זמן הוא חסכון במשאבים וזמן יישום.

כל משימה, המוטלת על מח' מיכון בסביבת שתף-זמן תסתיים במועד מוקדם בהרבה מאשר בסביבת BATCH (ואפילו בסיוע מסופי R.J.E).

ננתח לדוגמא מחזור חיים של כתיבת תכנית מחשב:

א. תרשים זרימה (או כל תיעוד אחר);

ב. קידוד
ג. ניקוב
ד. אימות ניקוב
} מספר סבבים

ה. COMPILE
ו. תיקון שגיאות
ז. העברה לספריה
} מספר סבבים

ח. ניפוי
ט. תיקון
י. עדכון ספריות
} מספר סבבים

יא. גמר.

תרשים הזרימה המצורף ממחיש מורכבות "מחזור חיים" זה.

מחזור חיים זה, כמעט ואינו תורם ישירות לנושא התכנות מבחינת הזמן המושקע בו. קיימים מקרים רבים הטוענים כי בסיסה זו, ויהא התכניתן מטולב במספר תכניות – סה"כ הזמן המושקע בעבודת התכנות עצמה ברוב המקרים נוסף מ-20% מזמן היישום כולו.

"בזבוז" לכאורה זה בא שתף-הזמן לתקוף, ובכך להגדיל את תפוקת (PRODUCTIVITY) התכנות במסגרת נתונה.

נראה מספר השוואות של השפעת מערכות שתף זמן על תפוקה:

א. __בבריטניה__

	שיפור	שתף-זמן	BATCH
זמן תוכניתן	61%	74.5 ימים	119.5 ימים
סה"כ משך יישום	85%	104.5 ימים	192.5 ימים

ב. __בארה"ב ב-T & AT__

	שתף-זמן	BATCH
שורות קידוד ליום	35.3	21.8
מס. COMPILES ליום	2.12	0.4
מס. ריצות DEBUG ליום	2.88	2.3
ריצת COMPIL נקיה ראשונה	2.00 ימים	2.5 ימים

ממוצע שיפור משוקלל – 62%.

בעבודה בשתף-זמנים – בצורה אינטרקטיבית יש רציפות בעבודה, ואין כל גורם אשר יעכב את התכנות בשלבים השונים להוציא תקלות במחשב.

פרט להגדלת התפוקה בגין רציפות העבודה מספקות רוב מערכות שתף זמן שירותים נוספים, אותם ניתן לסווג ל-2 סוגים:

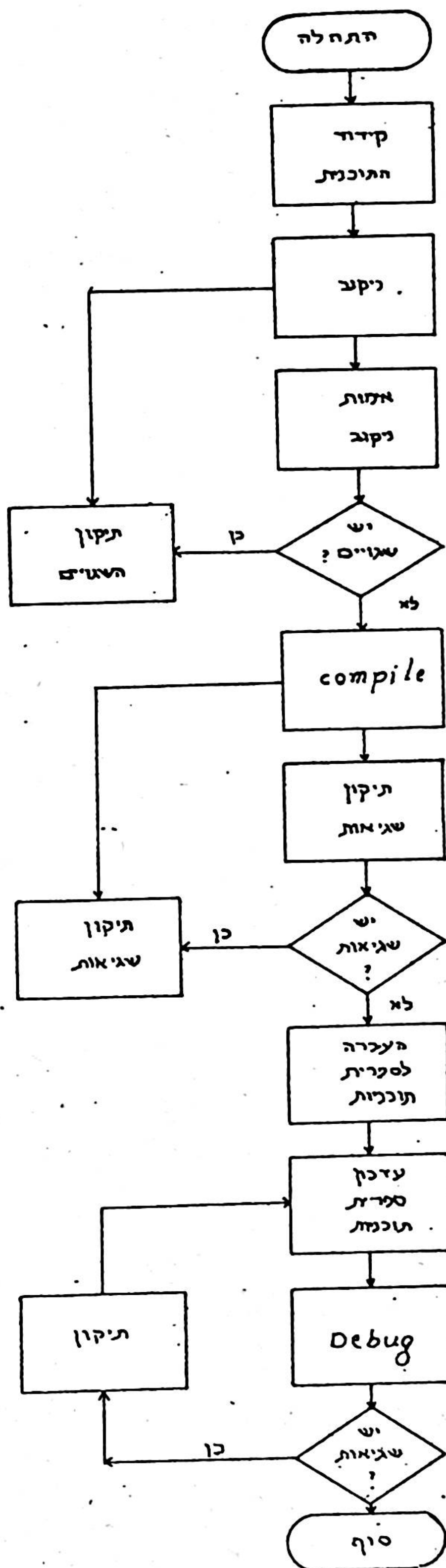
א. שפות מיוחדות;

ב. שירותים כלליים.

לרשות המשתמשים סדרה של שפות אינטרפרטיביות (APL וה-BASIC לדוגמא) אשר אינן מחייבות ההליך של הידור (COMPILATION). בשפות אלו, כתיבת תכניות ובמיוחד תהליך הניפוי נעשה קל לעין שיעור למול שפות העוברות הידור.

שפת ה-APL זוכה לתהילה רבה, ונבנו מספר מכונות חיקוב הפועלות בהתאמה לשפה זו, וכנראה, ששפה זו בצורתה הזו ובתוספת שיפורים תכסה גם את נושאי עיבוד הנתונים.

קיימות שפות אחרות, אשר לוורסיות שלהן יעילות מיוחדת בסביבת שתף זמן. דוגמא בולטת היא שפת ה-PL/1 אשר תחת T.S.O ובשימוש ב-CHECKOUT COMPILER ניתן ל"חולל פלאים". (במקרה זה את חוסר היעילות בגין קלות התכנות ב-CHECKOUT פותח ה-OPTIMIZER).


התחלה
קידוד
התוכנית
ניקוב
אימות
ניקוב
יש
שגויים ?
כן
תיקון
השגויים
לא
compile
תיקון
שגיאות
יש
שגיאות
?
כן
תיקון
שגיאות
לא
העברה
לספרית
תוכניות
עדכון
ספרית
תוכנית
Debug
תיקון
יש
שגיאות
?
כן
סוף

גם לשפות המוכרות בסביבת ה-BATCH כמו ASSEMBLER, COBOL, FORTRAN נותנות מערכות שתף הזמן שירותים מגוונים במיוחד פופולרית במערכות השונות שפת ה-FORTRAN.

במסגרת השירותים הכללים אשר מערכות שתף הזמן מעניקות למשתמשים בהן, סדרה ארוכה של פרוצדורות ופקודות (COMMANDS), בשלבי ה-EDITING, שלבי ה-CONPILATION וה-TEST.

<u>שתף זמנים וטכניקות בתכנות ותכנון</u>

עד כה נמנו התכונות החיוביות של מערכות שתף הזמנים ללא תלות בשיטות התכנות או העיבוד.

נושא שזוכה לפופולריות מרובה בעת האחרונה "טכניקות מתקדמות בתכנון ותכנות" מתגלה כמשתלב להפליא במערכות שתף זמנים. מתוך כל אוסף מרכיבי ה-I.P.T (IMPROVED PROGRAMMING TECHNOLOGY) נתעניין ב-2 במיוחד:

א. תכנות במבנים TTRUCTURED PROGRAMMING;

ב. תכנון מלמעלה למטה TOP-DOWN DEVELOPMENT.

תכנות במבנים הוא המשכו הטבעי של התכנות המודולרי. בתכנות במבנים אנו מניחים 4 צורות יסוד (רצף, ברירה, חזרה, בחירה) ומפרקים כל תכנית מחשב לקומבינציה ליניארית של 4 צורות אלו. כ"א מיסודות צורות אלו מהוה מודול או סגמנט (STUB) אשר לו יכולת קיום בדידה. גודלו של מודול כזה כ-50 פקודות או דף מחשב <u>אחד</u>.

את אוסף המודולים, אשר יכול להגיע לעשרות רבות, יש צורך לארגן בשיטה שתאפשר שליטה ויכולת ניפוי נוחים. לכך מצטרפת טכניקת TOP-DOWN DEVELOPMENT אשר מבטיחה כי בכל שלב במורד שלד המערך הכללי, כל המודולים הקודמים נשארים ללא שינוי ורק הרמה האחרונה במורד השלד הכללי עוברת ניפוי.

טכניקה זו, מבטיחה גם, כי ראשית לכל יש לבנות תכניות ראשיות ומובילות, ורק אח"כ את המודולים או השגרות הנקראות על ידן.

מה לשתף זמנים ולטכניקות בתכנון תכניות?

לאמיתו של דבר הפעלת התכנות במבנים והתכנון מלמעלה למעלה בשיטת ה-BATCH היא בבחינת לעג לרש.

התכניתן מטפל באופן כזה במספר רב של מודולים, אשר בכל רצון המפונים עליו, להפיק את מירב התועלת דורשים ממנו לעבוד עליהם במקביל. מאידך, כפוף הוא לכלים הקשוחים והחשובים לכשעצמם של התכנון בשיטת ה-TOP-DOWN, אשר אינו מאפשר ניפוי במקביל של תכניות אשר אינן ברמה זהה בשלד המבנה הכללי.

7/.

במקרים כאלו, ברוב המקרים תחכנה 2 פשרות או קומבינציה שלהן:

א. הקטנת מספר התכניות המסופלות ע"י בנית תכניות גדולות יותר (למעלה מדף לתוכנית או מודול);

ב. שבירת כללי התכנון בשיטת TOP-DOWN באמצעות טיפול במקביל במספר תכניות ברמות שונות.

כמובן, שהפעלת נוהלים קשוחים ובקרה מתמדת על ביצועם יכולה לאפשר יישום טכניקות אלו, במחיר אוירה לא נעימה במיוחד במחלקת המיכון.

אם נבחן באור זה את נושא שתף-זמנים, הרי שהוא מהווה פתרון אידיאלי לבעיות אלו.

לתוכניתן המנפה או כותב תכנית באמצעות המצג, נוח יותר אם התכנית לא תשתרע על פני יותר ממסך או שנים. במקרה זה התאחדו דרישות התכנות במבנים עם נוחיות העבודה בשתף זמנים.

בצורה בה מפורקת מערכת למת-תכניות כאלו, מסוגל תכניתן היושב מול מסך לסיים במהירות ועל נקלה מספר תכניות ב-SESSION אחד, ואז אינו מסתבך בכללי התכנון בשיטת ה-TOP-DOWN.

יתרון נוסף לטיפול בתכנית אחת בלבד, בכך שהתכניתן מתרכז כולו בבעיה, ואינו מתפצל בין מספר תכניות ובעיות, באופן שיוכל להפיק את מירב יכולתו.

המעניין הוא כי נושא שתף הזמן, והטכניקות המתקדמות בתכנון ותכנות פותחו במקביל, ורק עתה מבחינים בקשר שביניהן, וקיימות מספר מערכות שתף זמנים אשר מספקות שירותים ספציפיים למפעילים טכניקות אלו. (לדוגמא SPF STRUCTURED PROGRAMMING FACILITY ב-T.S.O. של חברת י.ב.מ.).

<u>שימושים נוספים למחלקת מיכון</u>

פרט לשימושים בשתוף זמן התורמים ישירות לתפוקת התכנות במחלקת מיכון, ניתן לייכס שימושים בשיתוף זמן ב-2 נושאים נוספים:

א. עבודת צוות ה-SYSTEM;

ב. מערכות אדמיניסטרטיביות במרכז המחשבים.

*

* ההתיחסות לאנשי ה-SYSTEM במחלקת המיכון היא אמביגיואית. מחד, ההם התוכניתנים ה יותר מיומנים המיצרים עבור מחלקת המיכון תכניות שירות והתקנות שונות הקשורות למערכות ההפעלה.

מאידך, הם המואשמים תדירות באי פעילות המחשב, או בהכנסת תיקונים אשר נועדו ל"כאורה" לשפר, אך גרמו לתקלות במערכת.

הייחוד בפעילות צוות ה-SYSTEM היא העובדה שכדי לטפל בנושאים שבאחריותם זקוקים לשחזור המצב בו ארעה התקלה, ועם איתור התקלה, לבצע את התיקון במהירות המידית שהרי המערכת משותקת וממתינה להפעלתה.

באמצעות שיתוף זמן ניתן לבצע הצצה במעין "מיקרוסקופ" לקרבי המערכת בעת ההתרחשות, ולתקן במהירות לאחר "הדיאגנוזה". אנשי צוות SYSTEM הפועלים בסביבת שיתוף זמן יגלו כי נושאים או תקלות אשר במערכות קונבנציונאליות היו פרובלמטיות, הופכות לבעיות קטנות ופשוטות. תיקון תקלות באמצעי הקלט/פלט, שחזור ספריות למיניהן, טיפול בקטלוגים, תיקון קבצים, טיפול בתקלות במערכות תקשורת, הופכות באמצעות מערכות שיתוף הזמן לפשוטות וברות פתרון מהיר.

* למרכז מחשבים בעל גודל בינוני ומעלה, בעיות ייצור בדומה למפעל יצרני. למרכז חמרי גלם-תכניות מחשב ונתונים, אמצעי ייצור -ציוד. מחשבים וציוד הקפי, קוי ייצור - מערכות מיכון שונות - לפנינו בעיה קלאסית של תכנון ייצור.

למרכז מחשבים אשר בעית תכניות העבודה היומיות והשבועיות, וקביעת העדיפויות היא חשובה, כדאי לנצל את משאבי שיתוף הזמן לאיסוף נתונים ועיבודים בזמן אמיתי.

ניתן להזרים תכניות עבודה חדשיות, שבועיות ויומיות ליחס להן עדיפיות, ולעדכנן באמצעות מערכות שיתוף זמן במשך יום העבודה, וליצור תכניות עבודה אלטרנטיביות. באותה מידה של קלות ניתן להמיר מערכות מעקב אחר תפוקת אנשי המיכון המבוצעת באופן ידני באמצעות מכשירי שיתוף הזמן הקיימים כבר. סדרה של נושאים המומלצים למעקב במסגרת ה-"INSTALLATION MANAGEMENT", ואשר אינם מבוצעים בגין צורך בניירת ופקידים החסרים לשם כך, ניתנים ליישום באמצעים פשוטים בסיוע מערכות שיתוף זמן.

יתרונות וחסרונות

יתרונות

א. <u>גמישות</u> - אפשרות לשימוש במשאבי מחשב בשיעור התואם את הצרכים הספציפיים של המשתמש, ולשם בדיוק עבור הצריכה.

ניתן לקבל פוטנציאל ציוד מתקדם מאד, אשר לא היה כדאי או אפשרי לרכישה ע"י המשתמש הבודד, אשר במקום להתקשר לציוד מתקדם היה רוכש מחשב קטן, או בכלל לא.

ב. <u>קלות בשימוש</u> - השימוש בשיתוף זמן קל מאד, הן להפעלה והן ללימוד. שפות תיכנות מיוחדות, ועזרים שונים של מערכות ההפעלה מפשטים תהליכים אשר הינם מורכבים בשיטות המקובלות. למשתתפים במחשב ביישומי שיתוף זמן שירותים מגוונים, והם אינם נזקקים לחיווך אנשי מיכון מקצועיים.

9/..

ג. <u>יחסי אדם-מכונה</u> - מערכות שיתוף זמן מאפשרות גישה אינטרקטיבית בין אדם ומחשב. משתמשים יכולים בשיטת ERROROTRIAL לנפות תכניות בעת כתיבתן, כשהמחשב בודק מדריך ומסייע בתהליך. דו-שיח כזה מאפשר פתרון של בעיות הנדסיות ומדעיות, ומספק למקבלי החלטות נתונים בשיטת היזון-חוזר וסיוע לתהליך קבלת ההחלטה.

ד. <u>קיצור זמן סבב</u> - מערכות שיתוף זמן מקצרות בהרבה את הזמן העובר מרגע הזנת הנתונים למחשב ועד הפצת תוצרת העיבוד. במערכות BATCH הזמן יקח שעות ולעיתים ימים, במערכות שתוף-זמנים התגובה מידית.

ה. <u>בחירת שפות</u> - רוב מספקי שירותי שתף זמן מספקים מגוון רחב של שפות תכנות, באופן שתנתן לכל מתמש האפשרות לשימוש בשפה ובנוהלים הנחוצים לו לפתרון בעיתו.

ו. <u>תוכניות שירות</u> - לרוב המשוקים המסחריים של מערכות שיתוף זמנים, פיתוחים עצמיים ונרכשים של תכניות שירות למיניהן, המאפשרות נוחות מירבית למשתמש, וחוסכים את הצורך בפיתוח עצמי.

ז. <u>רשתות תקשורת-ומסדי נתונים</u> - מספר חברות בארה"ב, ועתה גם באירופה מאפשרות התקשרות לרשתות תקשורת ארציות ובין ארציות, באופן שמתאפשר פיזור נתונים, וקליטתם במקומות התואמים את רשת השיווק או הייצור של הצרכן המסויים.

ח. <u>שירותים יעודיים</u> - מערכות רבות מספקות שירותים יעודיים למגוון רחב של משתמשים. לדוגמא: שירותים לבתי חולים, שיווק מכוניות, ביטוח ועוד. מערכות יעודיות נבנו במיוחד לנושאים אלו ואחרים, וייחודן בפשטות השימוש וההפעלה שלהן.

<u>חסרונות</u>

א. <u>אמינות</u> - במקרה של נפילה המערכת נשאר המשתמש ללא אמצעי מחשב. נושא הגיבוי למרכיביו השונים: המסופים, קוי תקשורת והמחשב המרכזי מקבלים משנה חשיבות במערכות שיתוף - זמנים.

ב. <u>איטיות קלט/פלט</u> - המסופים הישנים והקיימים הפועלים במהירויות של 10-15 תוים לשניה, עיכבו ופגמו בעבודה, ורוב החברות עוברות כיום למהירות מינימלית של 60 ואפילו 120 תוים לשניה.

ג. ניצולת מחשב נמוכה - מערכות הפיקוח על מערכי שיתוף זמן, גוזלות מרכיב נכבד מה-CPU והזכרון. במערכות אלו ניצולת המחשב היא נמוכה, והגדלת המשאבים לשם שיתוף זמנים מיקרת את מחירם.

ד. הגנה על נתונים - נושא אשר באחרונה זוכה להדים רבים הוא הגנה על נתונים. בעת האחרונה התפרסמו מספר מקרים של חדירה לבסיסי נתונים באמצעות מערכות שיתוף זמנים. ישנן אסכולות הטוענות כי בטכניקות הקיימות ברשתות התקשורת לא ניתן להבטיח הגנה מלאה על נתונים.

ה. עומס במערכת - מערכות מחשב המספקות שירותי שיתוף זמנים מאופיינות ע"י תנודות קיצוניות בעומס המערכת עד כדי שונות גבוהה כל כך שלשם הבטחת רמת שירות סבירה, יש צורך בפוטנציאל רב של ציוד אשר מיקר את מחיר השירות. גם כאשר הציוד רב, עשויים להיוצר "צווארי בקבוק". אשר עלולים לשתק את המערכת וליצור זמני תגובה אשר לא מאפשרים ללקוח לעבוד.

ו. מחירי תקשורת גבוהים - מחירי שכירות קוי טלפון הם גבוהים, ולעיתים נדרש הצרכן לשלם שכירות עבור קוים בקצב העברה גבוהה כאשר הנו מנצל למעשה מסוף בעל קצב עבודה נמוך.

ז. מחירי אחסון גבוהים לנתונים - רוב מערכות שיתוף זמנים דורשות נתונים בגישה מהירה, ולרוב על גבי אמצעי גישה ישירה, אשר מורכבים כמעט באופן קבוע ולפחות בשעות ההתקשרות, למחיר ממוצע של 0.5$ עבור 1K תוים, יאלץ צרכן אשר לו 10,000 רשימות בנות 80 טור לשלם כ-400$ לחודש.

למחיר זה יש להוסיף את מחיר אחסנת התוכניות.

השימושים של מערכת התקשורת במערך הלוגיסטי של צה"ל

סרן גדי אריאב , צה"ל

" הזווג בין הנדסת התקשורת להנדסת חרושת המחשבים היא אחת מהטכנולוגיות המסעירות של מאה מסעירה זו" - כתב ג'יימס מרטין בפתח אחד מספריו*.
העניין שמגלים היום בנושא זה סוגים שונים של ציבור, מאמצי המחקר, השימושים המתרבים והפתוח המתמיד הם "עדים" מהימנים לנכונות הערכה זו.

בטיפול בבעיות הכרוכות בטכנולוגיות מורכבות קיים הפיתוי להסיח את הדעת מהמטרה ולהתמקד באמצעים להגשמתה, כלומר לסטות מהבעיות אל הטכנולוגיה. הדבר נכון לגבי מחשבים בכלל ולגבי עבוד - רחק בפרט. חשוב, אם כך להדגיש את העובדה הטריוויאלית שחשיבותם של מרכיבי מערכת התקשורת נובעת מהתכלית שעבורה נוצרו, כלומר מהשימושים שלהם במערכת המידע.

במאמר זה תבחן, אם כן, מערכת התקשורת של המערך הלוגיסטי של צה"ל לאו דווקא מהצד הטכני הסהור שבה, למרות שבמהלך הפתוח נצבו בפני מקימי המערכת בעיות טכניות מורכבות ביותר כמו INFLIGHT CHECKPOINT/RESTART לעדכון ONLINE, ארגון שטחי הגלישה לקבצים בעדכון כזה והגישה אליהם, שלוב מחשבי מיני ברשת, הסבת תוכניות BATCH לתוכניות בסביבת תקשורת, MONITORING יעיל של הפעלת המודולים השונים ועוד - כיוצא בהן. הבעיות הטכניות ופתרונן מצדיקים ללא ספק סקירה נפרדת.

מערכת התקשורת של המערך הלוגיסטי בצה"ל היא אחת ממערכות התקשורת המופעלות בשיתוף זמנים במרכז המחשבים של צה"ל. מערכת זו מופעלת כבר מספר שנים כשמאפיין אותה קצב מואץ של התפתחות וזאת בשלושת התחומים:

- פריסת הרשת; מדובר היום ברשת בעלת עשרות מסופים.
- כמות התעבורה ברשת; כמות התשדורות אל המחשב בלבד היא בסדר גודל של עשרות אלפים ביום עבודה רגיל.
- מגוון השירותים והשימושים; התפתחות בשירותי הרשת מאופיינת בשני מימדים - ראשית שילוב של מערכת ורישומים נוספים במערכת התקשורת, ובמסגרת האפליקציות המשלובות - מעבר משימוש פאסיבי בראשית ימי המערכת - קרי: שאילתות ודווחים, לשימוש אקטיבי בהווה - עדכון ONLINE של מאגר הנתונים הלוגיסטי.

כאן אולי המקום לציין שפתוח המערכת לא נעשה על פי תוכנית אב אלא בסדר שהוכתב על פי "התפתחות טבעית" וצרכים שנתגלו תוך כדי ההפעלה. בפתוח התוכנה לתקשורת היתה המגמה לטפל בנושאים המרכזיים לפי סדר חשיבות יחסית (פחות או יותר) כשהטכניקה היא הסבת תוכניות ה-BATCH למבנה כזה שיאפשר הפעלתן גם בסביבת עבוד - רחק.

משמעותה הנוכחית של מערכת התקשורת למחשב ברשת המידע של המערך הלוגיסטי של צה"ל נתנת לאפיון בעזרת הטרמינולוגיה של BLUMENTHAL** באופן הבא:
מערכת זו משמשת כלי עבודה מאסיבי, יומיומי ושוטף של הבקרה בתפעולית (OPERATIONAL CONTROL) ולא רק משום כך, אלא גם באופן ישיר - כלי חשוב לשימוש של רובד הבקרה המנהלית (MANAGEMENT CONTROL).

*"TELECOMMUNICATION AND THE COMPUTER" , JAMES MARTIN PRENTICE-HALL, INC. ENGLEWOOD CLIFFS , N.J. 1969.

**"MANAGEMENT INFORMATION SYSTEMS - A FRAMEWORK FOR PLANNING AND DEVELOPMENT", SHERMAN C. BLUMENTHAL , PRENTICE-HALL , INC. ENGLEWOOD CLIFFS , N.J. 1969

הבעיה העיקרית הכרוכה בהצגת מערכת מורכבת כזו היא החלוקה המיתודית, כלומר גילוי אותו קריטריון חלוקה המאפשר הצגת המערכת כאוסף שלם וממצה של יחידות בדידות, במילים אחרות נחוצה כאן טקסונומיה. לצורך ההצגה סיגלתי את עקרון המודול כפי שהוא מופיע אצל בלומנטל כבסיס לחלוקה הפנימית בפירוט השימושים.

מודול (MODULE) הוא קטע תכנית, תכנית או קבוצת תוכניות שיש להן עניין משותף, המשרת פונקציה מסויימת בארגון ויש הצדקה - משיקולי עלות/תועלת - לקיומן העצמאי.
מימד שני בסווג הוא הרובד הארגוני במערך הלוגיסטי שאותו משמש המודול. בלומנטל מחלק את הארגון לשלושה רבדים עיקריים (לפי המודל של אנתוני) בעוד יש כאלה המוסיפים עליהם עוד רובד רביעי, הבסיסי מכולם.

הרבדים המקובלים הם:

(1) "רובד התנועות" ("TRANSACTION LEVEL") הוא הרובד שבו בערכת הפעילות הפיזית, מתבצעות התנועות עצמן, משם מגיעים הנתונים הבסיסיים, ולשם יש לבתב, למשל, דוחות שגויים. רמה זו נבלעת במודלים מסויימים ברמה הבאה.

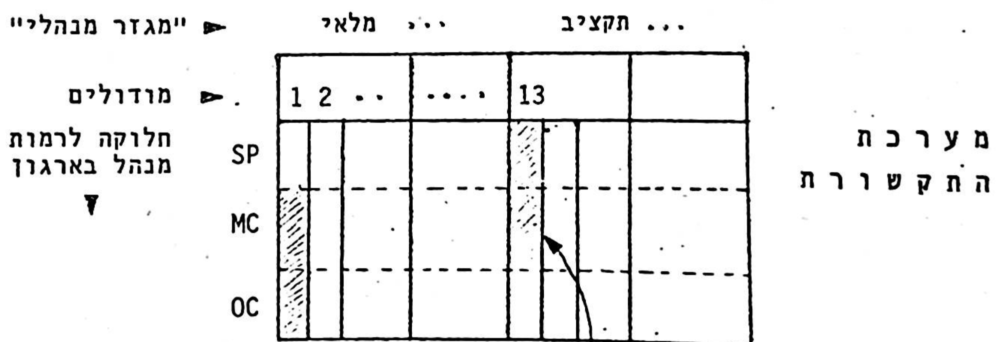
(2) "רובד הבקרה התפעולית" ("OPERATIONAL CONTROL") הוא הדרג המבצע, כלומר הרובד שבו בעשית פעולות הניהול האלמנטריות ולכן יש כאלה המכילים, כאמור, ברובד זה גם את "רובד התנועות", סוג שהוא לגיטימי לסוגים מסויימים של ארגונים, שבהם הפעילות הפיזית כרוכה במידה רבה של פעילות בנושא המידע (לדוגמה - ניהול מחסן). פעילות אופייבית לרובד זה היא הניפוק האוטומטי יש בו החלטות מובנות ("STRUCTURED").

(3) "רובד הבקרה המיבהלית" ("MANAGEMENT CONTROL") הוא הדרג המחליט, כלומר הרמה בארגון שבה מתקבלות החלטות שוטפות לגבי פעילותו של הארגון להגשמת מדיניות הארגון ויעדיו. סוג פעילות אופיינית לרמה זו היבה הבקרה התקציבית ואיתה הערכת היעילות בתפעול, וטיפול בנוהלי העבודה של הבקרה התפעולית.

(4) "רובד התכנון האסטרטגי" ("STRATEGIC PLANNING") הוא הרובד העליון שתפקידו - התוויית מדיניות וקביעת מטרות ויעדים לארגון כולו והתויית עקרונות פעילות למרכזי הבקרה המנהלית. פעילות אופיינית, במערך הלוגיסטי, לרובד זה היא למשל קביעת תקציבי הרכש הכולל והקצאת תקציבים כאלה.

מפתח החלוקה, אם כן, היבו דו מימדי, כאשר לצורך הבהירות כונסו כל הפונקציות המשרתות מגזר מנהלי אחד תחת כותרת אחת ("פונקציה תפעולית" בלשונו של בלומנטל - OPERATIONAL FUNCTION).

סכימת החלוקה היא:



מטבע הדברים תופסת בקרת המלאי מקום מרכזי בפעילות עבוד הנתונים של המערך הלוגיסטי. המודולים של מערכת התקשורת המסווגים תחת כותרת זו הם:

(1) קליטת תנועות מלאי - משרת את רמת ה-OC, ומאפשר הזרמת נתונים ממקום ההתרחשות ובדיקה מידית של סבירות תנועות אלו.

(2) עדכון ONLINE של נתוני מלאי משמש את הבקרה התפעולית בשני אופנים - בעתות רגיעה נובעת חשיבותו של מודול זה מהעובדה שהוא מאפשר למשתמשים להזרים תנועות חיוביות ולוודא במהירות את קליטתן הסופית והמלאה במערכת - לאחר השלמת בדיקות התקיבות בעת ביצוע העדכון עצמו. השימוש במודול לצורך הבטחת עדכניותו המירבית של מאגר הנתונים מקבל מימד וחשיבות מיוחדים בעתות חרום שבהן נערכות החלטות תחת לחץ זמן, ואמינות הנתונים ועדכניותם הם פרמטרים מכריעים לגבי איכות ההחלטות.

(3) דוחות קליטה ושגויים - מאפשר קבלת משוב על תנועות שעובדו במודול (1) או (2). מאפשר בזאת לאנשי הבקרה התפעולית להגיב על שגיאות הזרמה, לוודא קליטה ועדכון נאמנים. הדוחות נתנים באופנים ובחתכים הרצויים:

- כתגובה ספציפית לתנועה בודדת המוזרמת.

- דוחות כלליים הנתבים בחתכים רצויים (לפי נוחיות הטיפול).

(4) שאילתות בנושא מלאי - מודול השאילתות משרת את רובד הבקרה התפעולית ורובד הבקרה המנהלית כאחת. השימוש במודול הוא לבירורים ספציפיים במהלך תפעול המרכזים הלוגיסטיים ולקבלת הערכות מצב חלקיות לדרג הניהול.
השאילתות בתנות לפי דרישה במבנים שונים ובמגוון חתכים רב, המאפשר קבלת תשובה "עניינית" ועדכנית לשאילהה ספציפית, חלק מהשאילתות הוא העתק מדוייק של דוחות BTACH שהמשתמשים הורגלו אליהם במשך שנים, כשהאינפורמציה הכלולה בהם היא העדכנית ביותר.

(5) סימולציה של תהליך ניפוק - לפני ביצוע נפוק אוטומטי של מלאי מהמחסנים ליחידות בעשית סימולציה של הניפוק המיועד, המאפשרת לדרג הבקרה המנהלית לברר במדויק את השפעת מהלך זה על רמות המלאי השובות ולאמת את נחיצות הדרישה בהתאם לתגובות על הניפוקים הקודמים. מתוך כך בתנת לדרג זה הברירה להפעיל את התהליך בשיבוי מדיניות - חלוקה מסויימים, לבטל את מהלך הניפוק או ל"שחררו" לשלב הביצוע בפועל ללא הסתייגויות.

(6) ניפוק אוטומטי של מלאי - באמצעות מודול זה ניתן להפעיל מהלך של נפוק יזום של מלאי מהמחסים ליחידות, כאשר הניפוק מקבל כקלט גם דרישות לניפוק שהוזרמו ונצברו במערכת במשך פרק הזמן בין נפוק אחד לשני. המודול משמש את הבקרה התפעולית ומשחרר אותה מהצורך למלא באופן ידני שוברים וכמו כן מתבצע עדכון מלאים מתאים במאגר הנתונים (אם אמנם הוחלט במהלך הניפוק על ביצוע פעולת ניפוק).
במקביל לעבוד והפקת השוברים (=הוראת לנפוק של מלאי) במחסנים המתאימים מופק עבור מחסנים אלה מידע לאתור הפריטים במחסן. מידע אחר המופק במקביל הינו פירוט לצרכנים לגבי מה ש"מגיע" להם, כלומר שהוצאה עבורו הוראת ניפוק. מידע זה מעבר ברשת התקשורת ישירות אל ה"נמענים".

(7) מעקב אינדיבידואלי אחר פריטים - מדובר בפריטים בעלי ערך כספי ניכר המצדיק מעקב אינדיבידואלי (למשל, כלי רכב). מודול זה משרת בעיקר את הבקרה התפעולית לצורך בירור בעיות ספציפיות המתעוררות, למשל, בעת ביהול משק הרכב.

פעילות "קלאסית" בוספת של מרכז עיבוד הנתונים במערך הלוגיסטי היא בתחום ניהול הייצור או ליתר דיוק בבקרת הייצור, מאחר ואין המערכת עוסקת בתזמון (SCHEDULING) של משימות ייצור.

המודולים בנושא זה -
(8) (9) ו-(10) במקביל למודולים (1), (2) ו-(3) בנושא בקרת המלאי מאפשרים מודולים אלה הזרמת תנועות למאגר נתוני המשימות והייצור תוך קבלת תגובה מיידית לתנועות מוזרמות ואפשרות עדכון מיידי של נתונים, כמו למשל פתיחת משימות וסגירתן.

(11) מערך שאילתות ודווחים מאפשר לדרגים השונים לקבל אינפורמציה מעודכנת לפי דרישה לגבי משימות ספציפיות במבנה דוחות המקובלים בדווח ה-BATCH בנושא. מידע אופייני - פריטים משתתפים במשימה, צריכת חומרים במשימה, עמידה בלוח זמנים מתוכנן...

(12) DATA COLLECTION השימוש במודול זה נעשה ע"י שני הדרגים הארגוניים כאחת. המודול מאפשר קליטה אמינה של השקעות כח אדם במשימות, כאשר הנתונים נצברים תוך כדי הפעילות הרגילה.
מבחינה תפעולית מספק מודול זה תמיכה בניהול בסיסי של כח האדם (העדרויות, איחורים, שכר...) ותמיכה במעקב כללי יותר:השקעות כח-אדם וגזירת מקדמי צריכה של משאב זה.

בנושא הבקרה התקציבית במערך הלוגיסטי של צה"ל מתמקד במעקב אחר צריכת התקציב בהשוואה למה שהוקצה בתחילת התקופה (ללא האספקט הנוסף של תכנון המאפיין מערכות אלו במסגרות אחרות) והמודולים הפעילים במערכת התקשורת הם:

(13) פתיחה ועדכון של סעיפים תקציביים - המודול משמש לאחזקה שוטפת ONLINE של מאגר נתוני התקציב ותיתכנה פעולות כמו פתיחת סעיפים תקציביים, גריעה והוספה של סכומים בסעיפים...

(14) מערך שאילתות תקציב המספק לרובד הבקרה המנהלית מידע לפי דרישה אודות יתרות בסעיפים תקציביים לפי שנים, ביצול התקציב...

ניהול ההרכשה הוא חלק אינטגרלי של בקרת המלאי מאחר והוא מהווה את הפעילות המסדירה את ערוץ המלאי למערכת, מחמת החשיבות הרבה של פונקציה תפעולית זו מיחדים לה מקום בפרד. במערכת התקשורת מופעלים המודולים הבאים:

(15) ספקים - מודול זה מאפשר לרובד התפעולי ולרובד המנהלי לקבל אינפורמציה ספציפית אודות ספק מסויים או לזהות בשלב ההזמנה או הקבלה את הספק המעורב ב"ארוע".

(16) דרישות והזמנות - במודול זה נעזר הדרג המחליט בפעילותו הרגילה לצורך מעקב אחרי תכולת הדרישות והוצאתן אל הגופים העוסקים ברכש, ומעקב אחר התקדמות הטיפול בדרישה והתקדמות האספקה. שימוש בפוץ - לחזוי כמות המלאי העומדת להיות מסופקת למערך הלוגיסטי בטווחי זמן שונים.

הערה: עדכון ONLINE של מאגר נתונים זה נמצא בשלבי פיתוח מתקדמים. לעדכון זה תהיה גם השפעה על הבקרה התקציבית מאחר ודרישות הנפתחות במערכת יוודאו קיום תקציב מתאים עבורן והשרייבה אותו בהתאם.

במסגרת התמיכה של מערכת התקשורת ב-תכנון התובלה מופעלים המודולים הבאים:

(17) קליטה ועדכון של דרישות - באמצעות מודול זה מוזרמות הזמנות ודרישות להקצאת אמצעי הובלה כפי שהן נצברות בפני אנשי הדרג המבצע את התובלה. הדרישות נבדקות מיידית ונצברות בהתאם.

(18) תכנון תובלה עוסק בעריכת המלצות לדרג הבקרה לגבי הקצאה אופטימלית של אמצעי התובלה. ההמלצות מודפסות במסוף המרכז הלוגיסטי המתאים בצורה של שוברי נסיעה המוכנים למסירה ולביצוע.

(19) שאילתות ודווח פרמטרי בא לאפשר לדרג הבקרה המנהלית (ובמידה מסויימת לבירורים של הדרג הנמוך יותר) לקבל הערכות ביחס לנצילות צי הרכב, הזמנות בודדות, צפי עומס בזמנים נתונים, ריכוז תקציבי, ריכוז הובלה לפי אזורים גיאוגרפיים, תחזית תעסוקה יומית...

ישום חדש יחסית מהווים מודולי ההוראה וההדרכה. מנקודות מבט מסויימות ניתן לראות ב"ניצנים" אלה בסיס למודולים של תכנון אך שימושם הנוכחי הוא כאמור בהדרכה.

מודול לדוגמא:

(20) תכנון סדנא: מודול זה מכיל סימולציה אינטראקטיבית של פעילות סדנא לתיקון כלי רכב, כאשר ניתן לשנות במשך הסימולציה את הקצאת המשאבים בסדנא כזאת ולחשוף בעזרת תוצאות הסימולציה את ההשפעה שיש להקצאה זו או אחרת על תפקות הסדנא.

שימוש חשוב של מערכת התקשורת הוא התמיכה בניפוי ואחזקת תוכניות, באמצעות עזרים לתכנות הכלולים במערכת. קשה לייחס תחום זה למערך הלוגיסטי באורח ספציפי, אך מאחר ופנקציית עבוד הנתונים הינה חלק בלתי נפרד מהמערך הלוגיסטי כולו יש לגיטימציה למנות את השימושים ה"פנימיים" של המערכת במאמר זה. כלי העזר שיפורטו להלן פותחו בחלקם במסגרת ה-SYSTEM הכללי לשרות כלל משתמשי התקשורת ובחלקם לבקרה ונפוי המערכות הלוגיסטיות בלבד.

השימושים:

- R.J.E - כלי זה זוכה לפופולריות גוברת והולכת. הוא מאפשר למשתמשים שיש ברשותם הציוד המתאים להפעיל תוכניות BATCH לפי הדרישות המתעוררות באופן ספציפי במהלך העבודה. הפלט המבוקש מתקבל בתחנה השולחת ובתהליך ההרצה אין כל התערבות מצד מרכז המחשבים. התוכניות מסופקות היום למשתמשים ע"י המרכז לעבוד הנתונים, אך המגמה היא להגיע למצב שבו האחראים על התחנות יוכלו לכתוב בעצמם, בשפות דווח עליות כמובן, את התוכניות הנחוצות להם.
- שאילתות לגבי תכולה ומבנה פיזיים של קבצים ספציפיים שיש אליהם גישה במערכת התקשורת.
- תיקונים ONLINE של תוכניות וטיפול בספריות של תוכניות.
- מעקב אחרי קבצי הבקרה של המערכת.

כהערה מסכמת ניתן לאמר שהבעיות העיקריות בהן נתונה מערכת התקשורת היום מבחינת השימוש הן שתיים:

ראשית - מערכת השליטה (MONITOR) ומערכת בקרת המלאי המרכזית עומדות לפני סוף מחזור חיים והעומס עליהן עולה לעיתים על פוטנציאל השרות שלהן. נדרש עצוב מחודש להתאימן לדרישות הביצוע והיקף הפעילות החזוי.

שנית - בולטת העובדה שאין הרובד המנהלי העליון - רמת התכנון האסטרטגי - משתמש בכלים של מערכת התקשורת, בכל מקרה לא באופן ישיר. שילוב רובד זה בציבור המשתמשים יעשה ע"י פתוח שירותי דווח גמישים ומשוכללים ושלוב סימולציה לתמיכה בתהליך עריכת החלטות בדרג המנהלי הגבוה.

משך הזמן בו מופעלת המערכת כבר מאפשר סכום של לקחים ומגמות בשימושים של מערכת התקשורת:

(1) נתן לקבוע שמערכת התקשורת ככלי ניהול, ביחוד ברמות הבקרה התפעולית, נקלטה היטב כשהדגש - כפי שניתן ללמוד מתוך סטטיסטיקות לגבי השימוש בפועל - הוא באופן מובהק על האפשרות להקטין את זמן התגובה לאירועים לוגיסטיים עד למינימום וזאת באמינות גבוהה מאשר בעדכון BATCH. כיום למעלה מ-75% של התשדורות ברשת הן לצורך עדכון מיידי.

(2) מרכיבי המערכת העיקריים - חמרה, תכנה והפעלה קשורים זה לזה ביחס של "טורי" ולכן טיב המערכת בפועל איננו עולה על טיב החוליה החלשה שבה. פוטנציאל רב במיוחד טמון במיומנות השימוש לש מפעילי התחנות - אם בסגל הזוטר העוסק בהפעלה עצמה ואם בסגל הבכיר בתחנות כ-"סוכני הפצה" לשירותים ויועצים למשתמשים.

(3) מערכת התקשורת הנוכחית מכשירה ומבטיחה תשתית מתאימה מבחינת חמרה והטיפול בה,מצד התוכנה ומבחינת המודעות הגוברת של המשתמשים למחשב בכלל ולאפשרות ניצול מערכת התקשורת בפרט. עדות חותכת למודעות זו היא הלחץ מצד חוג המשתמשים לאפשר להם הפעלה נרחבת יותר של שירותי R.J.E.

(4) ההתפתחות של הרשת הנוכחית בכוון רשתות המחשבים היא טבעית וכמעט מתבקשת מאליה. בקשר לכך מקבלת ההערה הקודמת חשיבות מיוחדת.

(5) ניתן לאמר שהחומרה מהווה את הבעיה העיקרית - החלק מהתקציב המוקצב לכך (כרגע יותר מחצי) מצריך סטיות נכרות מאופטימום הפרישה של הרשת וממילא פוגע בשימושים פוטנציאליים.
המערכת עדיין רחוקה מהמצב שבו יוקצה מסוף לשירות צמוד לכל מספר פקידים העוסקים בנושא מסויים או לכל מנהל מחסן בבסיסי האחסנה.

(6) אספקט בעייתי אחר של החומרה הוא ברשת התקשורת. טיב הרשת ירוד ושיעור התקלות הוא בסדר גודל של 15% מזמן הפעילות, כאשר זמן התגובה לתיקונים הוא ממושך. משמעות עובדה זו מחמירה ככל שהשימוש במערכת גדל והתלות בה מעמיקה - תנודות ברמת האמינות של הרשת מהוות גורם בולם להרחבת השימוש.

יש להעיר שאין אחידות ברמת השירות בין האזורים השונים בארץ ומסתבר, כמו כן ,ככל שקיבולת הקו גדילה יותר התקבלה בפועל איכות שידור והפעלה משופרים יותר. יתכן בשל העובדה שקוים אלה חדשים יותר. עובדה שצריכה להדאיג מתכננים של מערכת REALTIME העתידים להשתמש בקווים בעלי קבולת נמוכה.

לא ניתן במסגרת סקירה בהיקף הנתון, לספק תיאור ממצה ומעמיק של פעילות מערכת התקשורת. למרות זאת נעשה נסיון לרמוז על מידת החשיבות של מערכת זו בחיי היומיום של המערך הלוגיסטי ולתת מושג על רמת ההתפתחות שלה. במצב הנוכחי בו המסגרות המקצועיות הצבאיות הן מהמובילות בנושא ומהיותן בית אולפנא חשוב לכח האדם המקצועי בענף בארץ, הרי יש לכך משמעות רבה מעבר לתפעולו של המערך הלוגיסטי בלבד.

מערכת עבוד נתונים בשיתוף זמן

מוגש ע"י: אריה כרמל - ורמי פישלר
מ.ו.ר - מחשבים וארגון (1975) בע"מ

כ ל ל י

מערכת לעבוד הנתונים בזמן אמיתי המוצגת כאן פותחה ע"י חברת מחשבים וארגון לפי הזמנה לפתרון נושא חישוב האגרות ברשות הנמלים. המערכת נמסרה לרשות הנמלים בספטמבר 1975.

המערכת לחשוב האגרות בזמן אמיתי המוצגת כאן מהווה חלק ממערכת מקיפה יותר אשר תטפל במינהל השירותים הניתנים למטענים ואשר תכלול את הנושאים הבאים:

- רכוז השירותים שניתנו למטענים
- פרוט מרכיבי השרותים
- חישוב האגרות לגביה עבור השרותים
- הכנת חשבונית וחיוב לקוח
- מעקב שוטף ופקוח על לקוחות באשראי
- מערכת באצוות:
  - לתעוד השירותים
  - ריכוזים חשבונאיים
  - נתוחים סטטיסטיים.

מהפכה בשיטת העבודה של חשוב האגרות והכנת חשבוניות ברשות הנמלים

מערכת זו מהווה שנוי מהפכני בשיטת העבודה, בחישובי השירותים, בהכנת חשבוניות, ובחיוב לקוחות עבור שרותים למטענים הניתנים ע"י רשות הנמלים.

* קליטה של נתונים המוזרמים ע"י העובד המקצועי ישירות למערכת (עדכון הקבצים בזמן אמיתי). באחריותו ובפיקוחו הישיר של העובד המקצועי.

* התחלה וסיום של הטפול בלקוח ע"י פקיד אחד תוך פרק זמן מינימלי.

* מערכת מורכבת של בדיקות אוטומטיות המאפשרת אתור טעויות ואי החאמות תוך כדי הכנסת הנתונים והכנת החשבון.

  - בדיקות לוגיות בין הנתונים לבין עצמם.
  - בדיקות מול מאגרי נתונים בחתכים שונים.

* חשוב מחיר של האגרות (מערכת איתור וחישוב מורכבת מאוד הכוללת מאות אגרות שונות).

* אחידות ואמינות בחשובי האגרות תוך עמידה בקצב מהיר של שנויים במדיניות תעריפים.

* פיטום מוחלט של עבודת החשב עקב העובדה שהחשב מזין למערכת נתונים פיזיים בלבד, בעוד שהמערכת מטפלת בכל פעולות החישובים, עדכון הקבצים, הדפסת החשבוניות ואחסון הנתונים לשימושים נוספים.

<u>נתונים כלליים למערכת.</u>

* – סקרים ראשוניים ותיקי תכנון כלליים בוצעו ע"י מר ליבנת ומר ש. פרומר.

– התכנון של המערכת הממוכנת ושילובה בדרישות של מערכת העובדה בזמן אמיתי רוכז בידי מר א. כרמל מחברת מ.ו.ר - מחשבים וארגון (1975) בע"מ, בכתיבת התכניות השתתפו:

– רמי פישלר – מתכנת בכיר אחראי על פיתוח שיטות הגישה והקשר עם מערכת ה- TSØ.

– בתיה כרמל – מתכנתת בכירה.

– מאז סיום כתיבת המערכת הוכנסו תיקונים ושיפורים במערכת TSØ ע"י מ.ל.מ ו-י.ב.מ.

מר נ. היילברון מנהל יחידת המחשב בנמל חיפה, בעזרתו של מר אמירם היידקר שולבו תיקונים ושיפורים אלה במערכת, וכן פיתחו מספר רב של שגרות כלליות ב- TSØ שחלקן שולבו גם כן במערכת הנדונה.

– איש הקשר לתכנון ותכנות המערכת שדחף לקדום הנושא לאורך כל הדרך היה מר י. מוניסה.

* – המערכת מורכבת מכ-70 מודולים.

* – תכניות הגישה והקשר עם מערכת ה- TSØ נכתבו באסמבלר תכניות באפליקציה נכתבו בקובול.

סה"כ נכתבו כ-70 תכניות (כ- 20,000 פקודות) המטפלות בכ-25 קבצים פיזיים הקשורים למערכת.

* – החדרת המערכת היתה פשוטה והעובדים לא נתקלו בבעיות מיוחדות בלימודה ובהפעלתה הנסיונית.

* – זמן הטפול בחשבונית אחת כ- 2-3 דקות. משך זמן זה ארוך בעיקר עקב התגובה של מערכת TSØ. מתוכננים שינויים במערכת המכוונים להפעיס ככל האפשר את מספר הפעמים בו ממתין המשתמש לתגובת ה"מחשב"; וע"י כך לשיפור הזמן הכללי להכנת חשבונית.

ראשוניות בעבוד נתונים בזמן אמיתי בפקוחה של מערכת TSO של י.ב.מ.

פיתוח המערכת תוך שימוש במערכת לשיתוף זמן (TSO) דווקא הוכתב ע"י לשכת השירות המספקת את שירותי המחשב לרשות הנמלים.

בפתוח המערכת לפי תכתיב זה עמדנו בפני בעיות רבות וקשות שלא היה ידע לפתרונן, לא בארץ ולא בחו"ל. (רק לחברת סיקורסקי אירקרפט היה נסיון בפתרון בעיות דומות אך בקנה מידה מצומצם הרבה יותר).

להלן חלק מהבעיות:

* עדכון קבצי אב בזמן אמיתי ללא חשש מאבוד מידע בזמן תקלות תוכנה או חומרה.

* תאום זמני העדכון של קבצי אב מתחנות שונות המשתמשות באותם קבצים, ובאותן רשומות.

* זהוי הצרכן כאשר מספר צרכנים משתמשים באותן תוכניות באותו זמן.

* ספול במצג - הגנה מפני הכנסת נתונים של אדם שאינו מוסמך לכך.
  - ארגון המצג מבחינת קדום דפים ושורות בצורה נוחה לצרכן.

* הדפסה של נתונים בזמן אמיתי על מדפסת הקשורה למערכת.

פ ר ק 1 - מערכת עבוד נתונים

תחת מערכת TSO של י.ב.מ.

1. מהות המערכת.

- המערכת המוצגת להלן עובדת תחת פיקוחה של מערכת TSO של י.ב.מ. המערכת נותנת שרות לקהל הלקוחות של רשות הנמלים, אך בשינויים מתאימים יכולה המערכת לשרת גם מערכת בנקאית, מערכת מכירות, מערכת גביית מיסי מכס וכו׳.

- מערכת אפליקציה מהסוג הנ"ל שואפות להגיע למצב בו יהיה באפשרותן:

  - להתחיל ולסיים "עיסוקה" עם לקוח ביחידת זמן אחת, ללא צורך בטלטול הלקוח מפקיד אחד למשנהו, עד לסיום ההליכים הקשורים ב"עסקה" הנדונה.

  - בצוע רשום הנתונים של ה"עסקה" ובדיקת הנתונים מבחינת נכונותם, וכן בדיקתם מול קבצים ומול טבלאות (יתרות, זכאויות של הלקוח, אפשרות מתן השרות בתנאי "העסקה" וכו׳).

  - הפעלת נוסחאות ואלגוריתמים של החשוב ל"עסקה" תוך העזרות בטבלאות תעריפים, קבצי אב, וכו׳..., ותוך המנעות מהכנסת שגיאות לחישובים ולהכנת חשבונית העסקה.

  - עדכון הקבצים במערכת כך שבכל יחידת זמן ניתן יהיה לדעת בדיוק את מצב המערכת (מבחינת יתרות לפריסים, יתרות להנה"ח וכו׳), וכן להמשך עיבוד נתונים אלה לצרכים חשבונאיים וצרכים סטטיסטיים.

- המערכת המפורטת להלן נותנת כלים נוחים לפתוח מערכות אפליקציה מהאופי המוגדר לעיל.

- המערכת תומכת, בעזרת מערכת ה- TSO של י.ב.מ. ושגרת תאום מיוחדת, בהפעלה בו-זמנית של תכניות האפליקציה ממספר רב של מצגים.

- המערכת תומכת בהפעלה נוחה של מספר מערכות אפליקציה, וכן בהפעלה נוחה של חלקי מערכת, אוטומטית, או לפי דרישת המשתמש.

- בהמשך העבודה, מובאת הדגמה של מערכת אפליקציה העובדת תחת מערך המסגרת הנדון.
האפליקציה היא של הכנה חשבוניה ללקוחות עבור שרותי נמל כפי שהיא מתבצעת באופן נסיוני בנמל חיפה.

2. ארגון המערכת.

מערכת המסגרת מורכבת ממספר מרכיבים:

- מ פ ק ח ת:

שתפקידה העברת הפקוח לתכניות אפליקציה שונות, לפי סדר מסויים, קבוע מראש בתוך האפליקציה, או בהתאם לדרישת המשתמש, וכן העברת הקשר בין תכנית האפליקציה ולנתוני הפעלתה (המשתמש, החשבונית בה מטופל וכו').

- טבלת פקודות:

מכילה את "מילון" הקשר בין המשתמש למערכות האפליקציה בהן תומכת המערכת, וכן את סדר בצוע שגרות האפליקציה.

- תכנית תקשורת למצג:

שולטת על העברת מיד ממאגריא הנתונים המגנטיים למצג ו/או חזרה, בעזרת שגרות של מערכת ה-T.S.O.

- שגרות קלט/פלט:

מאפשרות גישה למספר קבצים, המאורגנים בגישה ישירה מיוחדת והמשמשים את מערכת האפליקציה.

- שגרת תאום:

מתאמת בין הפעולות השונות הנעשות בו-זמנית בקבצים, ע"י המשתמשים בתחנות.

להלן תרשים כללי של המערכת ופירוט המרכיבים הנ"ל.

2.1 המערכת המפקחת.

המערכת המפקחת מהווה את המרכיב המרכזי בהיותה מרכז העצבים השולט בכל מרכיבי המערכת.

המפקחת מורכב מסדרת מודולים ופרוצדורות של TSO המתחילים עם התחלת העבודה של המשתמש, (כניסה למערכת) דרך הפעולה השוטפת של המשתמש במשך היום ועד לסיום ה"ישיבה" של המשתמש ליד המצג.

עיקר תפקידי מערכת הפיקוח:

2.1.1 יצירת הקשר בין המשתמש למערכת:

- לאפשר למשתמש המורשה (סיסמה מתאימה) להכנס לעבודה שוטפת במערכת.
- ל"קשור" את המשתמש עם מערכת המרכיבים השונים במערכת, ספריות מתאימות, קבצים, מודולים מתאימים להמשך העבודה וכו'.
- ליצור סדרת פרמטרים הקבועים למשתמש הנדון ולדאוג להעבירם לאחר מכן לתכניות האפליקציה השונות. כגון מספר המשתמש, מספר ה"תחנה" הלוגית, ופרמטרים נוספים הקשורים למשתמש הנדון.

2.1.2 עבודה שוטפת משך היום:

- הרגום דרישת המשתמש למודול המתאים שיש לאתר בספרית התכניות (שם הוא "חבר") ולהכניס מודול זה לפעולה.
- הודעה למשתמש על הצלחת העברת הפקוח למודול הנדרש (תכנית האפליקציה שנדרשה) והעברת הפקוח לתכנית זו.

- העברת מיצדים (פרמטרים) למודול אותו מפעילה התכנית המפקחת.
  - מיצדים כלליים הקבועים למשתמש הנדון.
  - מיצדים משתנים שאותם העביר המשתמש יחד עם דרישתו להפעלת האפליקציה המסויימת.
- קבלת הפקוח לאחר סיום פעולתה של התכנית שהופעלה.
  - הודעה למשתמש על סיום הבצוע של המודול הקודם.
  - ציון קוד חזרה עמו חזרה תכנית האפליקציה (המציין את האופן בו הסתיים בצוע התכנית).
- החלטה להמשך הפעולה.
  - לפי קוד החזרה הנ"ל (היו שגיאות, לא היו שגיאות וכו').
  - לפי סדר המודולים המוגדרים באפליקציה הנדונה בטבלת הפקודות.
  - התערבות המשתמש, והחלטתו על המשך הפעולה.
- סיום העבודה.
  - החזרת פקוח ל- TSO ולמעשה סגירת המצב אוטומטית לאחר שעות העבודה ולאחר משך זמן ללא טפול במצב.
  - סגירת מצב לפי דרישת משתמש והעברת פקוח ל- TSO.

2.2 <u>טבלת הפקודות.</u>

- טבלה זו מהווה את אחד המרכיבים החשובים במערכת המפקחת היא מוכנסת ע"י המערכת המפקחת ב"תחילת היום" (בתחילת העבודה של המשתמש עם המצג), ונשארת בזכרון כל משך הישיבה של המשתמש.

- הטבלה מהווה מילון קשר בין המשתמש לבין המודולים השונים העומדים לרשות המערכת כולה (תכניות האפליקציה השונות), המשתמש מחפש אפליקציה והתכנית המפקחת "מדפדפת" במילון זה לצורך איתור המודול המתאים לדרישת המשתמש.

- התכנית מפקחת איננה מסתפקת באיתור המודול הנדרש ע"י המשתמש. היא שבה לטבלת הפקודות לאחר סיום פעולת המודול הנ"ל ולפי הנתונים בטבלה היא תמשיך את פעולתה לפי האפשרויות הבאות:

  - לסיים את האפליקציה, לשוב למשתמש ולהמתין לדרישתו הבאה תוך הודעה מתאימה.

  - לעבור "אוטומטית" להפעלת מודול נוסף ללא תנאים.

  - לעבור לבצוע מודול נוסף בתנאי שהמודול שסיים זה עתה חזר עם קוד חזרה מסויים (סיום בדיקות בהצלחה, הכוונה לבצוע המודול הבא לפי נתוני המודול הקודם וכו׳) או לעבור לבצוע אחר אם החזרה מהמודול הקודם היתה עם קוד חזרה אחר.

  - לעבור לבצוע מודול נוסף בתנאי שהמודול שסיים זה עתה חזר עם קוד חזרה מסויים, אך זאת רק לאחר אישור נוסף של המשתמש (לא אוטומטי), לעבור לבצוע מודול אחר אם היה קוד חזרה שונה ולאחר אישור נוסף של המשתמש.

  - לעזוב את האפליקציה הנדונה ולשוב למשתמש להמתנה לדרישתו הבאה, ולא להמשיך כלל בבצוע יתרת המודולים המוגדרים לאפליקציה הנדונה (אם נדרש לקוח שכלל איננו קיים, או ~~שהלקוח~~ שהמשתמש החליט מרצונו לסגור את הטפול בלקוח זה למרות שלא סיים את הטיפול בו וכו׳).

2.3 תכנית קשר למצג.

תכנית קשר למצג היא התכנית היחידה המבצעת פעולות קלט/פלט עם המצג. התכנית עושה זאת בעזרת מכללים ושגרות של מערכת TSØ של י.ב.מ. המפקחת על פעולת המערכת כולה.

תכנית הקשר למצג פועלת עם מצג 3270 של י.ב.מ. אך היות וזוהי התכנית היחידה במערכת קשר עם המצג, אזי שנוי מערכת המסגרת לפעולה עם מצג מסוג אחר כרוך רק בשנוי תכנית זו בהתאם לדרישות של המצג האחר.
תכנית הקשר למסך בנויה מ-4 חלקים נפרדים:

- קריאת המסך - תוך כדי חפוש פקודת משתמש עבור המפקחת (פקודת משתמש מאופיינת ע"י סימן ≠ בתחילתה).
- כתיבת הודעה של המקפחת על המצג - כתיבה זו מורכב מכתיבת כל תוכן המצג תוך כדי שנוי השורה התחתונה. שורה זו משתמש אך ורק להודעות המפקחת.
- קריאת שורה אחת מהמצג - השורה שתקרא היא השורה שהוכנסו בה שינויים מזמן כתיבת המצג האחרונה. שורה זו תועבר לתכנית האפליקציה. בנוסף לשורה הנ"ל תעביר תכנית הקשר למצג לתכנית האפליקציה גם תו אחד מתחילת שורה במצג. אם תו כזה הוכנס ע"י המפעיל מזמן כתיבת המסך האחרונה. תו זה מהווה בדרך כלל קוד לתכנית האפליקציה על הצורה שבה רוצה המפעיל להמשיך בעבודתו (משמש כתחליף לכפתורי ה- Program Function, ומאפשר בצורה זו את עבודת מערכת האפליקציה גם על מצגים שבהם לא קיימים כפתורים אלה).
- כתיבת מצג - כתיבת המסך תעשה תוך כדי שנוי חלק משורות המצג או כל שורות המצג, בהתאם לדרישת תכנית האפליקציה (בכל מקרה השורה התחתונה במצג שמורה להודעות המפקחת, ולתכנית האפליקציה אין שליטה עליה).

2.4 שגרות קלט/פלט.

2.4.1 סוגי הקבצים.

על מנת לאפשר פתוח נוח של מערכות האפליקציה, פותחה שיטת גישה ישירה, המאפשרת גישה מהירה לרשומות שונות בקבצים. השגרות השייכות לשיטת הגישה נמצאות בזיכרון במשך כל זמן עבודת המערכת, ומאפשרת עדכון, שליפה של רשומות בקבצים מסוגים שונים, הקבצים הנתמכים ע"י שיטת הגישה הישירה המיוחדת הם:

- קבצים בעלי רשומות באורך קבוע, וללא נגררות.
- קבצים בעלי רשומות באורך קבוע, שלהן נגררות באורך קבוע, המשורשרות ביניהן ולרשומות המוביל. שיטת הגישה תומכת בשני סוגים שונים של נגררות כאלה.
- קבצים בעלי רשומות באורך משתנה.

כל סוגי קבצים אלה ניתנים גם לקריאה סדרתית רגילה הן ע"י תכניות האפליקציה, והן ע"י תכניות אחרות שאינן עובדות דווקא תחת מערכת המסגרת.

הערה: יש לציין ששיטת הגישה שתפורט להלן פותחה על ידינו בתחילת 1974 (התכנון היה ב-1973).
כיום ידועות שיטות גישה מקבילות שפותחו ע"י י.ב.מ. ויושמו לאחרונה בארץ (VSAM), אך בתקופה הנדונה הן לא היו ידועות בארץ ובארה"ב הן היו בשלבי נסוי ראשוניים.

2.4.2 שיטת הגישה לקבצים.

שיטת הגישה הישירה נעזרת בקובץ עזר היכול לשמש את שיטת הגישה לפעולה עם מספר בלתי מוגבל של קבצי נתונים מסוגים שונים. קובץ העזר (שיקרא מעתה קובץ האינדקס) בנוי מבלוקים פיזיים (גדולים יחסית) בגודל קבוע.

קיימים שני סוגים של בלוקים פיזיים בקובץ האינדקס:

- בלוקים של ספריה
- בלוקים של נתונים
- הבלוקים של הספריה נמצאים ברובם בתחילת הקובץ (נשמר שם מקום לעשרים בלוקים כאלה. (אם שיטת הגישה זקוקה לבלוקים נוספים מסוג זה, הם ירשמו במקום הפנוי הראשון בקובץ).

  בלוקים אלה מכילים כניסות שכל אחת מהן מצביעה לבלוק נתונים מסויים, ומכילה את המפתח הגדול ביותר והקטן ביותר הקיימים באותו בלוק נתונים (ממפתח מספר ..... עד מפתח מספר..... נמצא בבלוק מספר .....).

  הבלוקים של הספריה עצמם משורשרים זה לזה ע"י מצביע הנמצא בתחילתם.
- בלוק הנתונים מכיל רשומות קצרות כשבכל רשומה מוגדר האלמנט הנדרש וזהוי המצביע על רשימת הנתונים בקובץ הנתונים המתאים (קובץ לקוחות, קובץ מצהרים, קובץ ת. שער וכו').
- בבלוק הנתונים של קובץ האינדקס ניתן לרשום בנוסף למפתחות שהוזכרו לעיל, גם נתונים על האלמנט הנדון, דבר זה מאפשר:

  - אחסון אינפורמציה מענינת בקשר לאלמנט הנדון באינדקס עצמו ללא צורך בקובץ אפליקציה מיוחד לנושא. (וע"י כך חוסכים את הצורך בגישה לקובץ הנתונים).
  - איתור מאפיין של האלמנט הנדון:

    לדוגמא: אתור מספר הלקוח לו שייך המטען שמספרו....

    במקרה וקיים דווח של מס' המטען בלבד ורוצים לחייב את הלקוח המתאים.

להלן ההשלכה של הכניסות לקובץ נתונים כל שהוא של קובץ האינדקס:

גישה ישירה לרשומה דרך קובץ המפתחות

	כניסה נוספת	כניסה נוספת	מצוי בבלוק מס...	עד מפתח מס...	ממפתח מס...

בלוק ספריה בקובץ אינדקס

	כניסה נוספת	כניסה נוספת	נמצא בבלוק מס...	מפתח מס...

בלוק נתונים בקובץ אינדקס

	מפתח	

קובץ נתונים

א. הוספה.

הוספת רשומה בקובץ הנתונים גוררת לשרשרת הפעולות הבאה בקובץ האינדקס.

- הבלוקים של הספריה נסרקים, עד שנמצאה כניסת ספריה המצביעה לבלוק הנתונים האמור להכיל את הכניסה החדשה.

- בלוק הנתונים האמור נסרק לבדיקה האם כניסה בעלת מפתח זהה כבר קיימת.

- במידה והכניסה איננה קיימת, נבדק האם יש מקום פנוי בבלוק לעוד כניסה. אם וכן, מתווספת הכניסה לבלוק (ללא מיון) והבלוק נכתב חזרה במקומו. סדרת פעולות זו מסיימת במקרה זה את הוספת הרשומה.

- אם אין מקום לכניסה בבלוק (הבלוק מלא) ממויינות הכניסות בבלוק לפי מפתח (מיון פנימי בזכרון). הבלוק הממויין נכתב במקומו חזרה, וכמו כן נכתב בלוק חדש המכיל חצי מרשומות הבלוק הממוין (החצי בעל המפתחות הגבוהים יותר).
הכניסה בספריה שהצביעה קודם לבלוק המלא מפוצלת לשניים, כאשר כל חצי מצביע לאחד משני הבלוקים שנוצרו, עם כתיבת בלוק הספריה המכיל את שתי הכניסות, ניתן להתחיל בעדכון מחדש, וכעת יהיה מקום בכל אחד משני הבלוקים.. יש לציין כי במידה והתוכנית הסורקת (ראה לעיל) מוצאת כניסה בבלוק הנתונים, שאינה בגבולות המצויין בספריה, תמחק כניסה זו. כך שעם העדכון הראשון של בלוק הנתונים ה"ישן" (המיכל למעשה את כל הכניסות ממויינות), ימחקו כל הכניסות השייכות לבלוק הנתונים החדש.

- אם אין מקום לפיצול כניסת הספריה כך ששתי הכניסות החדשות תהיינה באותו בלוק, מוכנסת הכניסה המצביעה לבלוק החדש, לבלוק ספריה חדש, הנכתב בקובץ, ורק אח"כ נכתב הבלוק המכיל את הכניסה המצביעה על בלוק הנתונים הישן.

- יש לשים לב כי בשיטת עבודה זו לא קיים מצב שבו הקובץ "מעודכן למחצה". אם קורת תקלה במשך העדכון יתכן כי בלוק מסויים בקובץ האינדקס ישאר ללא שימוש וללא התיחסות אליו ע"י האינדקס (בלוק צף בקובץ), אולם מבחינה לוגית הקובץ ישאר תקין.

תרשים של שלבים בהוספת כניסה לקובץ האינדקס

1. מצב ראשוני

בלוק ספריה

בבלוק מס... 13	עד מפתח 'יד'	ממפתח 'יא'

בלוק נתונים 13 בקובץ האינדקס

שטח ריק	מפתח 'יג'	מפתח 'יא'	מפתח 'יד'	מפתח 'בב'	מפתח 'יבז'	מפתח 'יב'

השלבים הבאים מתיחסים למקרה שבו אין מקום בבלוק הנתונים לכניסה נוספת.

2. מיון בלוק הנתונים בזיכרון, וכתיבתו.

בבלוק מס... 13	עד מפתח 'יד'	ממפתח 'יא'

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יב2'	מפתח 'יבז'	מפתח 'יב'	מפתח 'יא'

3. חציית בלוק הנתונים הישן לחצי וכתיבת הבלוק החדש.

בבלוק מס... 13	עד מפתח 'יד'	ממפתח 'יא'

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יב2'	מפתח 'יבז'	מפתח 'יב'	מפתח 'יא'

בלוק נתונים חדש, מלא למחצה.

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יב2'

כתיבת בלוק הספריה עם הכניסה החדשה.

בבלוק מס.. 41	עד מפתח 'יד'	ממפתח 'יבב'	בבלוק מס 43	עד מפתח 'יבב'	ממפתח 'יא'

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יבב'	מפתח 'יבו'	מפתח 'יב'	מפתח 'יא'

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יבב'

מצב סופי אחרי עדכון. הכניסה המעודכנת בעלת מפתח אב.

בבלוק מס...49	עד מפתח 'יד'	ממפתח 'יבב'	בבלוק מס..31	עד מפתח 'יבב'	ממפתח 'יא'

שטח ריק	מפתח 'יאב'	מפתח 'יבו'	מפתח 'יב'	מפתח 'יא'

שטח ריק	מפתח 'יד'	מפתח 'יג'	מפתח 'יבב'

ב. **החלפה.**

החלפת מיקומה או תוכנה של רשומת הנתונים באחד מקובצי האפליקציה הקשורים לקובץ האינדקס מצריכה החלפת תוכן הנתונים (ההצבעה) בקובץ האינדקס עצמו פעולה זו מבוצעת ע"י:

- אתור הכניסה המקורית בבלוק הנתונים של קובץ האינדקס.
- החלפת תוכן הכניסה בתוכן חדש, וכתיבת בלוק הנתונים חזרה במקומו.

ג. **שליפה.**

שליפת רשומה מקובץ נתונים הקשור לאינדקס מתבצעת מחייבת אתור הכניסה באינדקס ולפי ההצבעה באינדקס, גישה ישירה לרשומה בקובץ הנתונים המתאים.

ד. **בטול.**

בטול רשומה מקובץ נתונים הקשור לאינדקס מבוצעת ע"י ביטול הכניסה בקובץ האינדקס המצביעה על הרשומה בקובץ הנתונים, בטול זה יעשה ע"י:

- אתור הכניסה הנדונה בבלוק הנתונים שבקובץ האינדקס.
- כל הכניסות הנמצאות בבלוק הנתונים, פיזית אחרי הכניסה הנ"ל, מוזזות כך שיכסו על הכניסה שאותה אנו רוצים לבטל. לאחר פעולה זו משתחרר בבלוק הנתונים שטח היכול להכיל כניסה נוספת, והכניסה שאותה רצינו לבטל - נעלמת.
- בקובץ הנתונים נשארת הרשומה ללא שינוי אך ללא הצבעה אליה ע"י האינדקס, כך שלמעשה זו "רשומה צפה".

2.4.3 **עדכון מתחנות שונות.**

תכנית המסגרת תומכת בהפעלה בו-זמנית מתחנות רבות. עובדה זו מצריכה תכנית שתתאם את עדכון הקבצים השונים, כך שכאשר תחנה מסויימת נמצאת בתהליך עדכון קובץ/אלמנט בקובץ (ותהליך זה יכול להיות מורכב ממספר פעולות קלט/פלט), לא תוכל אף תחנה אחרת לגשת לעדכון אותו קובץ אלמנט בקובץ. (אם לא היתה קיימת שגרה כזו, יכול היה עדכון בן-זמני של קובץ אחד משתי תחנות לגרום להריסה לוגית של אותו קובץ).

שגרת התאום מופעלת ע"י שגרות הקלט/פלט השונות. היא משתמשת במכללים DEQ, ENQ של מערכת ההפעלה DOS.

- יש לציין כי זמן עדכון הקבצים קצר כל כך ביחס לזמן העבודה הכללי במערכת מעין זו, המוגבלת למהירות העבודה של המשתמש, עד שהשפעת הפרעות אלה אינה מורגשת לגמרי מבחינת המפעיל.
- כ"כ יש לציין השיקולים שהנחו אותנו באלטרנטיבה שבין "סגירת" רשומה לבין סגירת קובץ במערכת, היתרון שזה נותן:
  - "סגירת קובץ" מצריכה פרמטר של הקובץ בלבד.
  - "סגירת רשומה בקובץ" מצריכה פרמטר שיבטא את הקובץ והאלמנט ה"סגור" בקובץ (מס' לקוח, מספר פריט וכו').

לעומת הקושי הנ"ל ברור שהשיטה השניה מאפשרת לשתי תחנות לעדכן בו-זמנית את שני אלמנטים שונים באותו קובץ.
האלטרנטיבה בין שני הנ"ל מוכתבת ע"י תדירות העדכון וזמני ההמתנה.

2.4.4 גיבוי לקבצים.

אחד היתרונות החשובים של השיטה שתוארה נובע מהעובדה שכמעט ואין צורך ב"שחזור" מסובך במקרה של תקלה במחשב.
השיטה מבוססת על כך שהעדכונים בקובצי האפליקציה אינם קשורים למערכת אלא לאחר כתיבת בלוק הנתונים של האינדקס, לפני פעולה זו (אם קיימת תקלה) לא בוצע כלל העדכון וסה"כ יכול להיות בלוק נתונים "צף" בקובץ האפליקציה והשחזור יהיה חזרה על בצוע כל תהליך העדכון. אחרי פעולה זו כל הקבצים במערכת מעודכנים.

תהליך השחזור לאחר תקלה פשוט, והאפשרות לשיבושים כתוצאה מתקלה אפסית.

נוסף על כך מספקת מערכת המסגרת שני אמצעים לבצוע שחזור מהיר:

- תכנית גבוי וכווץ היכולה להיות מופעלת בקלות לעיתים מזומנות. התכנית יוצרת מערכת חדשה של קבצים (תוך כדי עדכון כל ההצבעות בקובץ האינדקס כנדרש). הקבצים החדשים מכווצים (ללא כל השטחים המתים שנוצרו כתוצאה מביטול רשומות, או כתוצאה מתקלות תכנה במשך זמן העבודה), ומוכנים להמשך העבודה במערכת. מערכת הקבצים הישנה תשמש לגיבוי במקרה של תקלת חומרה.

- קובץ שיכיל את כל התנועות לפי סדר הגעתן, שהופעלו על הקבצים החל מזמן מסויים שנקבע ע"י התוכניתן (ראה קובץ גיבוי סעיף 5 (1)). במידה ואחד הקבצים נהרס כתוצאה מתקלת חמרה, כל מה שעל התכניתן או המשתמש האחראי לעשות, הוא לקחת את קבצי הגיבוי ולהפעיל עליהם את כל התנועות מהקובץ השומר אותם (הדבר יעשה ע"י תכנית מתאימה), ותוך זמן קצר ביותר מתקבלים הקבצים המעודכנים כפי שהיו לפני שארעה התקלה.

פרק 2 - מערכת לחשוב אגרות והכנת חשבונית עבור שרותי נמל

1. תאור כללי של המערכת הארגונית.

- המערכת עוזרת לחשבים של רשות הנמלים בקליטה, בדיקה וחשוב אגרות עבור שרותים שנתנו למטענים תוך העזרות במסופים הקשורים למחשב מרכזי גדול (י.ב.מ. 370/168).
- היבואן מגיע לחשב עם טופס תעודת שער (אותו הוא מקבל מסוכן האניה).
- החשב מכין חשבונית ללקוח תוך בדיקות וחשובים מתאימים.
- היבואן משלם את הסכום המתאים ורק אז הוא יכול לגשת למחסני הנמל לשחרור מטעניו.
- להלן מובא פרוט השלוב של המערכת הממוכנת במערכת המסגרת המוגדרת לעיל.

2. שילוב המערכת בתכנית המסגרת.

מערכת החישובים שתוארה לעיל מבוצעת בעזרת 4 תכניות אפליקציה:

- תכנית דו שיח בין המשתמש לבין תוכנית קליטה ובדיקום.
- תכנית חישוב אגרות.
- תכנית עדכון הקבצים.
- תכנית הדפסת החשבונית.

תכנית המסגרת עוברת מאפליקציה אחת לשניה לפי סדר זה, אלא אם היתה התערבות המשתמש.

3. תכניות האפליקציה.

3.1 תכנית דו שיח בין המשתמש לבין תכנית קליטה ובדיקות.

3.1.1 תאור כללי.

- קליטת נתונים של תעודת שער באמצעות מצגים מדגם 3277 תוך הנחיות המשתמש במלוי תוכן השדות השונים.
- בדיקת הנתונים שהוזנו וסימון השגיאות שהתגלו.
- הצגת תעודת השער או חלקים ממנה לפי דרישת המשתמש (כולל סימול השגיאות שהתגלו בבדיקות).
- מתן אפשרות להכניס שנויים/ביטולים בנתוני תעודת השער, בכל אחד משלבי הקליטה (גם לאחר חשוב אגרות).

3.1.2 ארגון המצג.

- ארגון המסך הנדון מוכתב מדרישות האפליקציה, המגבלות של TSO כפי שהיו ידועות בעת תכנון ותכנות המערכת, וכן מגבלות הציוד כפי שהוכתבו ע"י רשות הנמלים (מצגים ללא PF וכו').
- רשומת תעודת השער מורכבת ממוביל ומנגררות המתחלקות לשני סוגים:
  - נגררות אגרות סבלות והגדרת המטענים.
  - נגררות לאגרות רציף, אחסנה ושונות.

בהתאם לכך יחולק המצג לשלושה אזורים עיקריים:

- אזור המראה את שדות המוביל.
- אזור המראה את השדות השייכים לאגרות הסבלות. (כל נגררת סבלות תתפוס שורת מצג).
- אזור המראה את השדות השייכים לאגרות הרציף, האחסנה, והאגרות השונות (גם כאן כל נגררת תתפוס שורת מצג).

אם יש מספר רב של נגררות, יופיע חלק של נגררות הסבלות ביחד עם המוביל של תעודת השער, ואילו ההמשך (שאין לו מקום במצב הראשון), יופיע במצג נוסף לפי בקשת המשתמש.
בכל שלושת האזורים שצויינו קיים תת אזור המכיל את קודי השגיאות אם יש כאלה בנתונים. תת-אזור זה מופיע בצד שמאל של המסך.
שורות 21-22 - תת אזור זה מכיל גם את תרגום קודי השגיאות ("נומריות", "לא נמצא בטבלה", "סבירות תאריך" וכו').

3.1.3 <u>מצב קליטה ומצב הצגה.</u>

תכנית הדו-שיח יכולה להמצא באחד משני מצבים:

- מצב הצגה בו מציגה התכנית את תעודת השער, והמשתמש יכול לשנות או לבטל של נתון ו/או לבטל שורה או שורות בתעודה (ביטול שורה זהה עם ביטול הנגררת המתאימה בתעודה).
- מצב קליטה - בו יכול המשתמש להוסיף תעודת שער חדשה ו/או שורה או שורות נוספות בתעודת שער קיימת. (הוספת שורה בתעודת שער זהה להוספת נגררת מתאימה).
- המעבר ממצב קליטה למצב הצגה ולהפך, וכן הנחיות שונות אחרות של המשתמש לתכנית הדו-שיח, מבוצעות ע"י הכנסת קוד מיוחד בטור 1 של שורה כלשהיא במצג.
  המשתמש יכול לדרוש בצורה זו את הפונקציות הבאות מהתכנית:
  - הצגת תעודת שער (מצג ראשון)
  - הצגת המצג הבא - אם תעודת השער משתרעת על יותר ממצג אחד
  - בצוע בדיקות תקינות על הנתונים שהוזנו ע"י המשתמש
  - הוספת שורה מכל סוג שהוא
  - בטול/שנוי שורה מסויימת
  - סיום התכנית ללא כתיבת התעודה המעודכנת בקובץ תעודות השער
  - סיום התכנית עם כתיבת התעודה המעודכנת בקובץ תעודות השער
  - בטול התעודה הנדונה בכללותה
- סיום התכנית וכתיבת תעודת השער המעודכנת בנתונים שהזין המשתמש בקובץ תעודות השער (ראה תאור קבצים). תעודה זו תקרא מקובץ זה ע"י תכניות האפליקציה הבאות כאשר הן תופעלנה, לשם המשך העבוד.

```
ערך סיפ ס-א פריקה עד.. מ.. מצהר   יעד יבואן עמיל אנכ ת. שער .
.-------+-+-+--+--+-+----+--++--+---+-+-+-+--+--+-+----+-+------+---+
0000001 212 08128 01111  05  00018 0001 0088 140276 1     22112
.
שגיאות סכום חיוב תעריף תאריך סכום חיוב  ק נסמן  שעור אמדן ק נסמן שעור.
                 הוצאה    אחסנה    ו אחסנה  אחוז ימים ו רציף  אחוז.
.+-+--+-----+---+--++--+--+--+-----+---------------+-------+--+---+--------+------
100  020118 1  20  100  032201 1       110.57  160375    22.10     4355.00    04*!
.
שגיאות  סכום חיוב תעריף ק נסמן שעור רהמ אר מס מ אריזות קוד משקל מספר .
        ו סבלות אחוז  זה וך כ יחידות משקל מטען  סודר .
.+----+-------+++-------+-+-+-+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+------------
00011      1000 01      100 1 41 10  100 100 020118 1  100.00  1,122.20
::::פ //////// :://////// : // :: //: ::: ////// : ////.// ::,::::.::
. .
שגיאות  סכום חיוב תעריף ק נסמן שעור אמדן תח מס מרכיב קוד מרכיב מספר .
        ו אחוז ימים נה וך כמות 2 משקל כמות 1 סודר .
.+----+-----+++-------+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+------
00012      2000 01      100 41 10  20   100 031110 1   30.00    4553.35
::::פ //////// :://////// :: //  ::   ::: ////// : ////.// :::::::.::

04  תאריך לא סביר

א1010# בפעולה על 0000001     ג8 21# בהמתנה על 0000002
```

3.2 תכנית חישוב אגרות.

- חישוב אגרות שונות – עבור כל נגררת לגבי הנתונים הכמותיים שלה, וכן חישוב אגרות-לנתונים כמותיים מצטברים של כל תעודת השער.
- איתור מצבים חריגים ושגיאות בקשר שבין סעיפי החיוב לנתוני התעודה.
- בדיקות אשראי של הלקוח כפי שהוא מופיע בקובץ הלקוחות בהתחשב ביתרה ובאומדני חיוב לא סופיים של הלקוח.
- לאחר סיום שלב זה, יש ברשומת תעודת השער בקובץ תעודות השער ערכי החיוב כפי שחושבו עבור השרותים השונים וכן ציון החריגים והשגיאות שהתגלו תוך כדי מעבר זה.

3.3 תכנית עדכון הקבצים.

- הכניסה לעדכון הקבצים מתאפשרת רק אם לא היו שגיאות חמורות בתעודת השער הנדונה בשלבים קודמים, וכן רק אם התעודה עברה חשוב אגרות והיה אישור מתאים של המשתמש.
- חשוב לציין, שכל עדכון הקבצים באפליקציה זו נעשה יחד במסגרת תכנית אפליקציה אחת וזאת על מנת למנוע כל אי התאמה בין הקבצים.
- התכנית תכתבה על מנת לרכז עדכון הקבצים במודול אחד שאיננו מחייב התערבות המשתמש בין עדכון קובץ אחד למשנהו – וע"י כך לקצר את משך זמן בצוע העדכון וכן להקטין את הסתברות התקלה בזמן פעולת תכנית זו.
- כ"כ כדאי לציין שבגלל הנ"ל, כל תקלה בזמן העבודה איננה משמעותית לגבי המערכת אלא אם כן היתה בזמן עדכון הקבצים, כך שנושא "שחזור" מצטמצם לתכנית זו בלבד.

- הקבצים המעודכנים ע"י תכנית זו הם:

  - קובץ לקוחות
  - קובץ מצהרים
  - קובץ רשומונים (משולב בקובץ האינדקס).
  - קובץ תעודות שער
  - מודול זה כולל גם אופציה להפעלה לאחר תקלה שארעה באמצע טפול בתעודת שער. במקרה זה מאתרת התכנית את השלב שבו ארעה התקלה, ואם השלב הוא בזמן עדכון קבצים תאתר התכנית את המקום שבו ארעה התקלה (נעשית בדיקה לקובץ הראשון שאיננו מעודכן בצורה נכונה וסופית), והעדכון ממשיך מהמקום בו ארעה התקלה.

3.4 הכנית הדפסת החשבונית.

- הכניסה לתכנית זו מתאפשרת רק אם התעודה עברה את שלב עדכון הקבצים.
- מטרת התכנית היא להדפיס חשבונית עבור הלקוח על מדפסת הנמצאה בצמוד למצג.
- ניתן להדפיס אותה התעודה מספר פעמים, כאשר בכל ההדפסות הבאות יירשם "העתק" לציון העובדה שהמדובר הוא בהעתק של החשבונית.

4. הפעלה שוטפת של המערכת.

4.1 תכנית פיצול קובץ תעודות שער.

- מערכת ה- ON-LINE של חשוב האגרות לרשות הנמלים היא חלק ממערכת גדולה יותר המשולבת בעבודה יומית BATCH תוך עבוד קבצי המערכת שנוצרו ב- ON-LINE.

- עם סיום העבודה היומית, נלקח קובץ תעודות השער, (המהווה קובץ עבודה למערכת ה- ON-LINE), ותכנית מיוחדת מחלקת את הרשומות שבו למספר קבצים:

  - קובץ תעודות סגורות - שהוצאה להם השבונית.
  - תעודות שיש בהן שגיאות והנמצאות עדיין במצב דו-שיח עם המשתמש.
  - תעודות שער בעבודה שעברו את שלב הדו-שיח עם המפעיל, אולם הקבצים עדיין לא עודכנו בנתונים השייכים להן, או שצפויים בהן עדיין שנויים.

- כל אחד מקבצים אלה מועבר לתכניות שונות של המערך הכללי לשם עיבוד נוסף, קובץ תעודות שער מכווץ ונקי יועמד לרשות מערכת ה- ON-LINE של היום הבא.

- ההחלטה להעביר תעודות שער ממערכת ה- ON-LINE לקבצים שיטופלו בהמשך באצוות היא לפי פרמטרים שינתונו לתכנית. פרמטרים אלה יגדירו:
"העברת כל התעודות מסוג ..... שהטיפול האחרון בהן היה לפני התאריך...".
בצורה זו ההחלטה ארגונית תקבע מתי תעודות שער גמורות יוצאו ממערכת ה- ON-LINE ומתי תעודות שער שגויות יוצאו ממערכת זו ללא אפשרות גישה אליהם במערכת ה- ON-LINE.

4.2 הפעלת המערכת ממצגים שונים.

- לכל משתמש קיים חבר בספריה מיוחדת בה יש סדרת פקודות המועברות למערכת TSO והמהוות בסיס להפעלת המערכת ע"י המשתמש הנדון.
- הגישה ל"חבר" זה היא ע"י קריאה ל-" LOGON " מתאים, כך שהפעלת הנושא פשוטה מאוד.
- המערכת פועלת בעזרת תכניות אחידות לכל המשתמשים הנטענות לבצוע מספריה אחת ויחידה, וכן משתמשת המערכת בסדרת קבצים משותפת לכל המשתמשים.

ההבדל בין משתמש אחד למישנהו הוא במיצדים הנמצאים בספריה המוזכרת לעיל והמאפיינים את המשתמש המסויים.

קבוצת דיון מס׳ 11:

"תכנה בסיסית ומערכות הפעלה"

יו"ר ע. גרינגרד

נ. אחיטוב

מ. איצקוביץ

ניתוח מתמטי-כלכלי של אגרגציות של תרשימי זרימה

ניב אחיטוב
הפאקולטה לניהול
אוניברסיטת תל-אביב

1. רקע

תרשים הזרימה הינו כלי לצורך תאור הקשרים הלוגיים בין מרכיבים שונים של מערכת מידע. זוהי הצגה ציורית-סימבולית של המרכיבים במערכת.

תרשים זרימה יכול לשמש לתאור תכנית בודדת או לתאור מהלך המורכב מכמה תכניות או לתאור מערכת מידע המורכבת מרשת של מהלכים. האלמנט הבסיסי של התרשים הוא המלבן (בפועל - סוגי פעולות שונים מקבלים סימבולים שונים אך אנו נניח תאור אחיד ע"י מלבן). מלבן יכול לייצג פקודה בודדת, רוטינה, לולאה שלמה (loop), תכנית, מהלך, תת-מערכת וכו׳, הכל לפי עניינו של המשתמש במידע. מלבנים יכולים לייצג גם קבצים למיניהם.

2. תחום הדיון

אנו נתייחס רק לחלק הדינמי של התרשים המלבני (תרשים זרימה), דהיינו לאותו חלק שמתאר תכניות ומהלכים. נתעלם ממלבנים המתארים מרכיבים סטאטיים, דהיינו קבצים, דוחות, רשומות קלט וכיו"ב.

המרכיב האלמנטרי יהיה פקודת מחשב בודדת, אשר ניתנת כאמור לייצוג ע"י מלבן. קבוצת הפקודות של תכנית תסומן ב- C.

$$C = \{C_1,\ldots,C_n\}$$

באשר C_i - פקודה בתכנית ספציפית.

הערה: אבר בקבוצה מאופיין ע"י מהות הפקודה וע"י מיקומה בתכנית. לכן - פקודה זהה שמופיעה במקומות שונים באותה תכנית, כל מופע שלה ייחשב כאבר שונה.

מספר פקודות המקובצות יחדיו למלבן אחד תיקראנה רוטינה. קבוצת רוטינות של תכנית תסומן ב- R .

$$R = \{R_1,\ldots,R_m\}$$

קיימת פונקצית מיפוי (נוסח אפנדיקס B מתוך איג׳ירי [1]) מ- C על R.

$$r : C \xrightarrow{onto} R$$

מחלקת אקויולנציה ב- C הינן כל הפקודות המתקבצות לאותה רוטינה ב- R.

אנו נגדיר שאם	$r_1 : C \longrightarrow R^1$
ואם	$r_2 : C \longrightarrow R^2$
ואם	$R^1 \neq R^2$
אזי	$r_1 \neq r_2$

הגדרה: תהיינה $r_1 \neq r_2$ פונקציות מיפוי מ- C ל- R^1 ול- R^2 . r_1 תיקרא יותר עדינה מ- r_2 אם כל אבר במחלקות האקויולנציה של C לגבי r_1 מוכל או שווה לאבר במחלקות האקויולנציה של C לגבי r_2 . (ראה איג'ירי אפנדיקס B [1])

מחלקת אקויולנציה של C C_i לגבי פונקציה r מסומן ב- $[C_i]_r$.

משפט מס. 1 בנספח B של איג'ירי [1] יקבל את המשמעות הבאה:

אם $r_1 : C \to R^1$ הינה הקבצה של פקודות לרוטינות ואם גם $r_2 : C \to R^2$ הינה הקבצה כזו, אזי ניתן לבצע הקבצה מ- R^1 ל- R^2 אם ורק אם r_1 יותר עדינה מ- r_2. (במקרה זה קיימת פונקציה : $r_3 : R^1 \xrightarrow{onto} R^2$, או בסימון שונה $r_3 : r_1(C) \xrightarrow{onto} r_2(C)$).

3. הקבצות תקפות

לב [2] מדבר בעבודתו על הקבצות "הגיוניות" מבחינה מאזנית. בחרשימי זרימה, מגבלות ההקבצה יותר חמורות: ראשית, קיים "הגיון" מסויים של הקבצת פקודות לסברוטינות ולרוטינות, הגיון אשר קשור בלוגיקה של התכנית. שנית, הקבצה יכולה להעשות רק על פקודות סמוכות זו לזו.

המגבלה השניה, משמעותה הינה שהקבוצה C איננה מכילה אוסף של אלמנטים בעלי סדר מקרי, אלא זו סדרה של אברים אשר בה כל אבר מקבל סיפרור על פי מקומו היחסי מתחילת התכנית.

r תיקרא הקבצה תקפה אם מתקיימים התנאים הבאים:

א. לכל $R_k \in R$, R_k הינו תמונה של מספר אברים עוקבים $C_j, C_{j+1}, \dots, C_{j+l}$ השייכים ל- C .

ב. r שומרת על הסדר, כלומר אם

$$R_k = r(C_j, \dots, C_{j+l})$$

$$R_{k'} = r(C_{j'}, \dots, C_{j'+l'})$$

ואם $k' > k$

אז $j' > j + l$

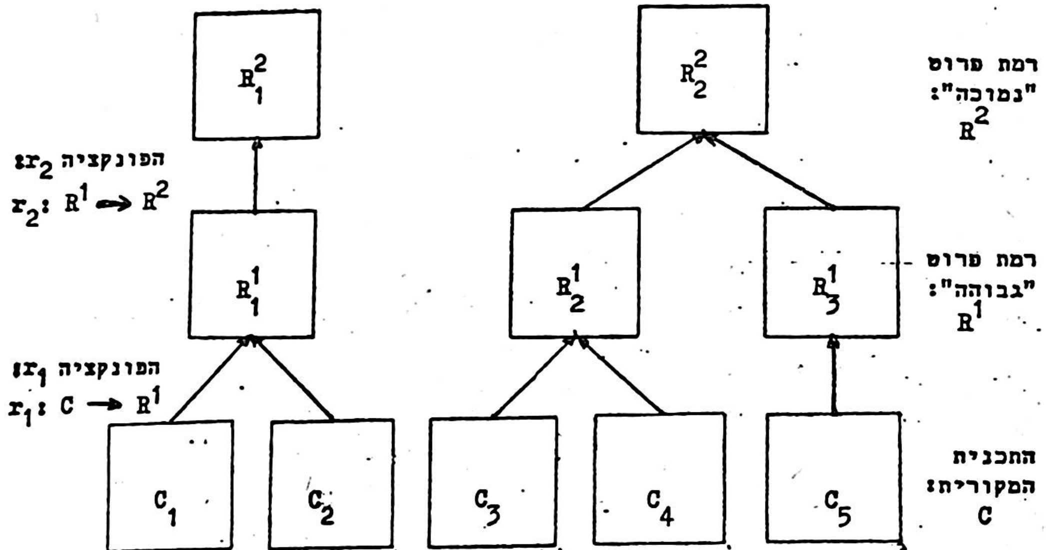
מכאן נובע שגם R הינה סדרה.

המגבלה הראשונה העוסקת בהגיון התיכנותי של ההקבצה, קשה להצגה פורמלית בשלב זה. ניתן לבצע אלימינציה של הקבצות לא הגיוניות ע"י קביעת ערך של $-\infty$ למידע במלבנים מסוימים, אך על זה ידובר בהמשך.

4. היררכיה של הקבצות

המשתמש הבודד מבצע הקבצה של פקודות תכנית עד לרמת פרוט מסוימת שעונה על צרכיו (דיון על כך יובא בהמשך העבודה). אנו מניח שאם ירצה רמת פרוט יותר נמוכה (הקבצות יותר רחבות) הוא ימשיך לקבץ מתוך ההקבצה הקודמת. הנחה זו עקבית במדה מסוימת עם תנאי "ההגיון" של ההקבצה שמוזכר בסעיף הקודם.

הצגה גראפית:



בשרטוט שלעיל מתקיים ש- $R^1 = r_1(C)$

ו- $R^2 = r_2(R_1)$

וגם $R^2 = r_2(r_1(C))$

ברור ש- r_1 יותר עדינה מאשר הפונקציה המורכבת $r_2 * r_1$.

הצגה מתמטית: תהי נתונה סדרה (תכנית) C.

סדרת הפונקציות $r_1, \ldots, r_k$ תיקרא הקבצה היראכית כאשר מתקיים:

א.

$$r_1 : C \longrightarrow R^1$$

$$r_2 : R^1 \longrightarrow R^2$$

$$\vdots$$

$$r_k : R^{k-1} \longrightarrow R^k$$

ב. לכל j, $k > j \geq 1$ מתקיים ש-

$r_1 * r_2 \ldots * r_j$ יותר עדינה מאשר

$r_1 * r_2 \ldots * r_{j+1}$.

5. <u>ערך המידע של מלבנים</u>

עד כה הוצגו ההקבצות בצורה של פונקציות מתמטיות אך ללא התייחסות למשמעות הכלכלית שלהן. לכן עד כה, כל סדרה של הקבצות היראר‎כיות הינה תקפה. נרצה לצמצם את מספר ההקבצות התקפות וזאת ע"י מתן משמעות כלכלית לתרשים הזרימה.

תרשים הזרימה הינו מוצר מידע דהיינו אמצעי למסירת מידע. ניתן לכן להניח שיחידת יחידה של מוצר זה הינה יחידת מידע ולמוצר עצמו ישנו ערך שהוא ערך המידע.

פונקציית ערך המידע של תרשים מלבני איננה מונוטונית עולה ככל שעולה רמת הפירוט, אם לא כן, כל צרכן של המידע היה שואף לפרוט מכסימלי. עצם העובדה שצרכנים שונים שואפים למידת פרוט שונה מצביעה על כך שלכל אחד מהם פונקציית תועלת שונה והיא איננה תלויה מונוטונית אך ורק במידת הפרוט. (דיון מפורט על כך ייעשה בהמשך.)

בשלב זה נניח שכל צרכן קובע לעצמו מידת פרוט "אופטימלית" מבחינתו. מידת פרוט זאת מתבטאת בכמות המידע שבתרשים, קרי במספר המלבנים.

<u>טענה:</u> תהיה נתונה רמת פרוט אשר נקבעה ע"י צרכן המידע. אזי הצרכן יבחר באותה הקבצה היראר‎כית, אשר תביא אותו לרמת הפרוט הנתונה כך שערך המידע השולי של כל מלבן יהיה זהה לזה של כל מלבן אחר.

הוכחה: נניח שהטענה איננה נכונה. כלומר בהקבצה r_j כלשהי קבלנו מלבנים R_1^j ו- R_2^j כך שערך המידע השולי שלהם שונה, ונניח שערך המידע השולי של R_1^j גבוה מזה של R_2^j.

אזי כדאי ללקוח לגרוע פקודה (יחידת מידע) מ- R_2^j ולהעבירה ל- R_1^j. "מחיר" כל מלבן לא ישתנה וערך המידע יגדל; ובמונחים כלכליים: באותה עלות ייצור ניתן להגיע לערך יותר גבוה של התוצר.

<u>מסקנה:</u> תהיינה $H_1^l,\ldots,H_n^l$ כל ההקבצות ההיראר‎כיות (סדרות של $r_1,r_2,\ldots,r_k$ – ראה סעיף 4) שהן תקפות בהגדרת התקפות הטכנית (ראה סעיף 3). אזי הקבצה היראר‎כית תקפה גם מבחינה כלכלית אם היא מקיימת שעבור פונקציית ערך מידע מסוימת, ערך המידע השולי זהה בכל מלבן ומלבן.

מכאן שמתוך $H_1^l,\ldots,H_n^l$ קיימת רק קבוצה חלקית של הקבצות היראר‎כיות אשר תקפות מבחינה טכנית וכלכלית כאחד.

<u>הערות:</u>
1. הטענה שלעיל מסתמכת כמובן על הנחת רציפות של הקבצות. במציאות, הקבצה מרמת פרוט אחת לשניה איננה רציפה וגם העברה של פקודה ממלבן אחד למלבן שני איננה ניתנת להצגה אינפיניטיסימלית. לכן בפועל רושם תרשים הזרימה ישאף להגיע למצב של ערך מידע שווה בכל מלבן, אך יסטה מכך כתוצאה מחוסר רציפות.

2. אם נניח שערך המידע נמדד ע"י פונקציית אנטרופיה לוגריתמית (ראה [2]) אזי ההוכחה המתמטית של הטענה הינה פשוטה, כיון שעבור M מלבנים פונקציית האנטרופיה מקבלת ערך מכסימלי כאשר $H(R) = \log M$ וזה מתקיים כאשר ערך כל מלבן יהיה $H(R_i) = -\frac{1}{M}\log\frac{1}{M}$

3. ברור שהטענה שלעיל הינה אינטרפרטציה של כללי התפוקה השולית המקובלים בכלכלה.

6. יחידות ההקבצה ההיראַרכית

עד כה הגדרנו מהי הקבצה וצמצמנו את תחום הדיון תחילה להקבצות תקפות, אחר-כך להקבצות היראַרכיות ולבסוף להקבצות היראַרכיות תקפות גם מבחינה כלכלית. סעיף זה יציב טענה יותר חזקה ויצמצם את המבחר להקבצה היראַרכית יחידה לכל פונקצית תועלת.

טענה: יהי נתון צרכן מידע בעל פונקצית תועלת ידועה כלשהי. אזי קיימת סדרה היראַרכית של הקבצות (הקבצה היראַרכית מהצורה $r_1 * r_2 * \ldots * r_k$) אחת ויחידה אשר הינה אופטימלית עבורו, לכל דרגת פרוט.

הוכחה: הטענה מוכחת כמעט מיידית. בסעיף 3 הראינו שהקבצה תקפה הינה כזו ששומרת על סדר האברים העוקבים. בסעיף 4 הראינו שהקבצה היראַרכית היא הקבצה תקפה. בסעיף 5 הוכחנו שערך המידע השולי של כל מלבן זהה. מכאן שכאשר הצרכן מגדיר רמת פרוט נתונה (במונחים של מספר המלבנים שהוא רוצה לראות נגד עיניו), קיימת רק חלוקה אחת של התכנית הבסיסית אשר תקיים את התנאי של סעיף 5 תוך שמירה על הסדר העוקב שנדרש בסעיף 3.

נניח שהצרכן הגדיר רמת פרוט כלשהי - רמה מספר 1: אזי קיימת r_1 יחידה כך ש- $R^1 \longrightarrow C : r_1$ ו- R^1 אופטימלית מבחינה כלכלית.

נניח שיגדיר כעת רמת פרוט יותר נמוכה (תרשים יותר מקובץ). אזי קיימת $R^2 \longrightarrow R^1 : r_2$ אשר תפיק R^2 אופטימלית, וחוזר חלילה.

כלומר, עבור כל רמת פרוט ניתן לבצע אגרגציה ישירות מהתכנית הבסיסית C להקבצה אופטימלית ברמה זו או להגיע אליה דרך סדרה של הקבצות היראַרכיות אשר שומרת על כלל האופטימום לכל אורכה. סדרה זו היא יחידה, עבור צרכן יחיד.

7. מדוע מבצעים הקבצה?

אם נקבל את הגישה הפשטנית האומרת שכל הקבצה פרושה אובדן מידע, נחטא למציאות. במציאות אנשים מבצעים הקבצה, הן בתרשימי זרימה, הן בסטטיסטיקה (מדדי נטיה מרכזית ומדדי פיזור) והן בחשבונאות. יוצא איפוא שתועלת הצרכן מן המידע איננה רק פונקציה מונוטונית עולה של כמות המידע, אלא שהינה בעלת אחת התכונות הבאות (או שתיהן יחדיו):

א. קיים תחום בו הפונקציה הינה מונוטונית יורדת ולכן האופטימום איננו בפרוט המכסימלי

ו/או ב. ערך המידע תלוי במספר משתנים ולא רק בכמות המידע.

בסעיף הבא נתייחס לתכונה השניה.

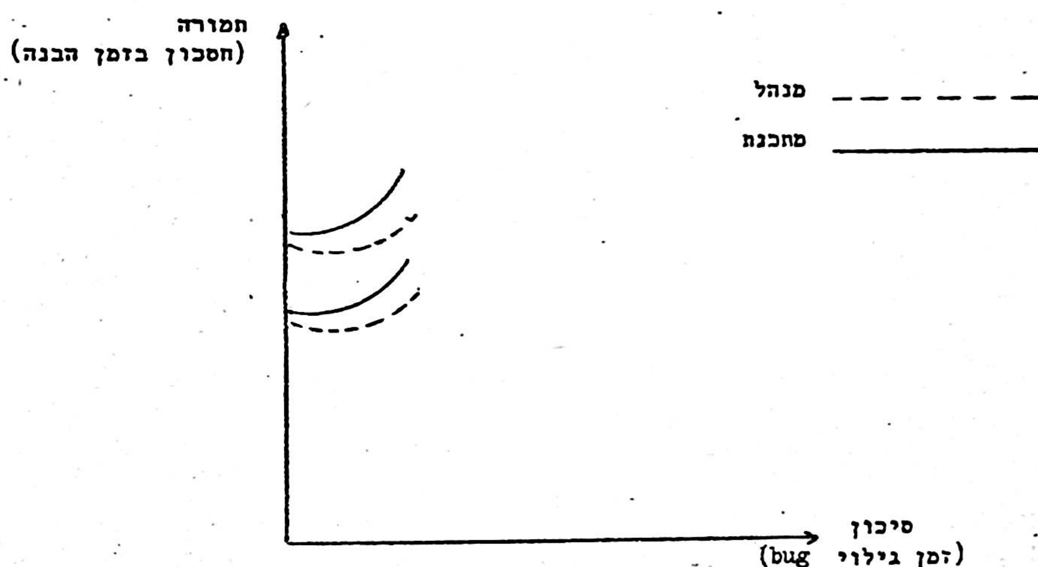
8. תועלת דו מימדית מתרשים זרימה

הקבצה של תכנית מקלה על הקורא להבין את מהות הנושא. "זמן ההבנה" הולך ומתקצר ככל שההקבצה פחות מפורטת. לכן קיימת איזושהי תמורה להקבצה, אשר הינה פונקציה מונוטונית יורדת ל"זמן ההבנה". ככל שהזמן מתארך, התמורה קטנה (פונקצית הפסד כתוצאה ממחיר הזמן).

מאידך, הקבצה גוררת עמה סיכון. ככל שהתרשים פחות מפורט, גדלה תוחלת משך הזמן שיידרש לתיקון bug בתכנית אם יימצא כזה.

לצרכן המידע קיימת משפחת עקומות שוות תועלת במישור שמימדיו הם תמורה וסיכון. צורת משפחת העקומות משתנה בהתאם לאופי הצרכן: מתכנת יעדיף פרוט (פחות סיכון) על "זמן הבנה" קצר. מנהל יעדיף תמורה על סיכון.

הדבר מוצג בשרטוט הבא:



עבור תמורה נתונה, המנהל מוכן לשאת ביתר סיכון מאשר המתכנת ולכן יש שוני במשפחת עקומות שוות התועלת שלהם.

9. <u>חזית יעילות: הקבצה אופטימלית</u>

<u>טענה:</u> הקבצה היראַרכית אופטימלית יוצרת חזית יעילות במישור סיכון/תמורה, כלומר אם

$$r_1 : C \rightarrow R^1$$

$$r_2 : R^1 \rightarrow R^2$$

$$\vdots$$

$$r_k : R^{k-1} \rightarrow R^k$$

ואם $r_1, r_2, \ldots, r_k$ אופטימלית, אזי לא ייתכן ש- R^i יהיה דומיננטי על R^j במישור סיכון/תמורה לכל i ו- j המקיימים $1 \leq i,j \leq k$.

<u>הוכחה:</u> יהיו R^i ו- R^j שני תרשימים השייכים לקבוצה $R^1,\ldots,R^k$ ונניח $i > j$.

R^i אופטימלי לרמת פרוט נמוכה יותר ו- R^j אופטימלי לרמת פרוט גבוהה יותר.

R^i יותר "מסובך" מ- R^j כי הוא מקובץ יותר. מאידך זמן ההבנה של R^i קצר יותר מזה של R^j ולכן התמורה ממנו גבוהה יותר.

לכן R^i אינו דומיננטי על R^j ו- R^j אינו דומיננטי על R^i.

מכאן ש- $R^1,\ldots,R^k$ יוצרת חזית יעילות.

<u>מסקנה:</u> לכל צרכן מידע ישנה הקבצה אחת אופטימלית שהינה נקודת ההשקה של חזית היעילות עם עקומה שוות תועלת שלו.

במונחים פרקטיים: קיים תרשים זרימה אחד ויחיד שהוא אופטימלי עבור אדם נתון. תרשים זה מאופיין בכך שערך המידע השולי בכל מלבן שלו הוא קבוע. התרשים מתאים לצרכי הקורא הספציפי מבחינת רמת הפרוט שלו כך שיהיה נוח להבנה אך גם נוח ל- debugging הכל לפי סוג השימוש הנעשה להרשים.

10. <u>השלכות מעשיות</u>

לדיון התאורטי שהוצג לעיל עשויות להיות גם השלכות מעשיות במידה והנושא יעבור הרחבה ועיבוד.

א. הניסוח המתמטי של אגרגציות יכול לשמש פתח דבר לפורמליזציה של תהליכי ההצגה בתרשים מלבני. כיום קיימות תכניות אשר מפיקות תרשימי זרימה מתוך שורות קידוד. הפקה זו היא מכאניסטית. אם תפותח גישה מתמטית בהסתמך על האמור לעיל, ניתן יהיה להציב פרמטרים אשר יגדירו את היקף ההקבצה של התכנית המשרטטת. במקרה כזה ניתן יהיה להשתמש באותה תכנית על מנת להפיק תרשימים מפורטים ופחות מפורטים, לפי עניינו של המשתמש.

ב. הניתוח הכלכלי של ערך המידע בהקבצות יכול לשמש בסיס לקביעת מדיניות הצגת תרשימי זרימה למשתמשים השונים. דבר זה ייעשה ע"י בנית טכניקה של שקלול לפרמטרים השונים. פרמטרים משוקללים אלו יהוו קלט לתכנית שרטוט של התרשים המלבני.

ברור שעדיין רב המרחק בין הניתוח הראשוני שבוצע כאן לבין יישום מעשי של הרעיון. לכן יש להתייחס לעבודה זו כאל סקיצה של גישה עתידית לנושא תיעוד התוכנה.

רשימה ביבליוגרפית

1. Ijiri, Y., The Foundations of Accounting Measurement, Prentice Hall, 1967.

2. Lev, B., Accounting and Information Theory, American Accounting Association, 1971.

INFLIB - A SOFTWARE INFORMATION-RETRIEVAL SYSTEM

MEIRA ITZKOVITZ

Technion - Israel Institute of Technology
Computer Center
Haifa, Israel

ABSTRACT

Efficient usage of the computer's resources requires the maximum utilization of standard programs. Using existing programs reduces reprogramming and saves programmers valuable time. However, locating an existing program is not always a simple matter. The amount of software available grows daily but the lack of a centralized library makes its location so difficult that it is often easier and faster to reprogram the entire project, thus wasting machine and human resources. Finding a program in a program library should be like finding a book in a public library. There should be catalogues, indexes and bibliographies available as aids in locating the desired program: It was with this goal in mind that the Technion Users' Group designed the INFLIB information retrieval system. This system, developed around a simple, easy to learn query language, requires no previous programming experience and allows the user to quickly access programs related to his needs. Through the use of keywords, the user may question the system on the most general to the most specific aspects of the program. The system has a broad base and is particularly designed for the programmer who has only a vague idea of what he needs. Such a user may request a list of all the programs available on a given subject. Having seen the list of programs available, he may then begin a process of elimination.

The system, which operates in both batch and interactive modes, is built on a main data base, two secondary datasets and a tracer data set which provides feedback information about the system's operation.

This information system represents the first serious attempt to computerize the subject of software organization. Such a system may have far reaching effects in the computer field as it will reduce duplication, thus making programming more efficient and saving computer time and manpower lost on reprogramming. The system's simple query language makes it easy to learn without limiting its flexibility and scope.

INTRODUCTION

Efficient usage of computer resources requires the

maximum utilization of standard programs, procedures and subroutines. This avoids the duplication and waste resulting from the reprogramming of common problems and algorithms. The amount of standard software available to the user increases daily. In fact, it would not be exaggerating to say that almost every problem facing today's programmers has been solved previously and has an appropriate solution existing for it. The process of redeveloping a program (coding, debugging, optimization, etc.) is unnecessary. It is far more efficient to use an available program.

Using an existing program may seem like a simple matter. However, the large amount of material available makes it difficult to locate the necessary program. The vast quantity of software and the lack of centralized documentation often make it preferable for the user to solve the problem alone starting from A-Z rather than try to locate an existing program which may meet his requirements. The problem becomes greater in a large computing center where many programs are available. There are many programming languages, a large number of users and no means of communication available for exchanging information. Finding a suitable subroutine for solving a specific problem becomes as complicated as reprogramming the entire routine alone. The lack of centralized documentation also makes it difficult to run a program even if the programmer knows the program name.

Locating a program in a computer program library is similar to locating a book in a technical library. Professional librarians use indexes, catalogues and bibliographies as tools for helping them locate information. However, in the computer field this subject has been greatly neglected. This may partly be due to the rapid growth in the computer science field itself. Originally, computer centers assigned their top level programmers to the job of librarian. From this the idea of a Users' Group, responsible for program maintenance, program consultation, program organization and software collection, developed. The problem here, however, is that these people lack professional training in library sciences and, therefore, rely heavily on their memory and knowledge of the system. As the field is continually growing this is not always sufficient. A systematic, efficient method of information retrieval is required.

It is for that reason the Technion Users' Group has developed the INFLIB system. The main objective of INFLIB is to organize all program information into a

central system. All the details of finding and operating a program have been concentrated into one computerized system which the user can access quickly and easily. The remainder of this paper explains the basic principles of INFLIB and gives the user some background in its usage.

USING AN EXISITING PROGRAM

While conducting research or working on a project, the user may encounter problems that requires the computer for its solution. A program is required which executes a particular set of operations and solves the problem. In order to take advantage of such a program the user must first know if it already exists. If it does, he must find out the following details about it.

a. In what programming language it is written?
b. What algorithm does it use?
c. Can it be used in an interactive environment?

If the user finds that a program is available and its general description seems to meet his needs, he must obtain further details such as:

a. What is the program name?
b. Where is it located (by program libraries)?
c. What type of routine is it (main program, subroutine, program package, etc.)?
d. What job control cards, data preparation and parameters (type and precision) are required for operation?
e. What is the output?
f. With which compiler can it be used?
g. How much memory space is required?

More information may still be required after knowing the answers to the above questions. If this is the case, the user should refer to other resources such as mathematical literature, literature describing the algorithm, detailed program documentation, the consultant or the program's author.

OPERATING PRINCIPLES OF THE INFLIB SYSTEM

The section above gave a brief description of the basic information required by each user before he can effectively use a program. The INFLIB Information System was designed with this in mind. INFLIB provides a simple logical way for the user to interrogate the system until he finds a suitable program. The user may ask all of the questions above or select a limited

number of them. Queries can be made in any order. For example, the user may know the program name and want to know only the details concerning the operating procedure. It is also possible that a user knows the best algorithm for solving a problem and wants to locate the program which uses it. Most programmers tend to favor a particular programming language or even a particular compiler. INFLIB accommodates the user by informing him of the programming language as well as the compiler used by a specific program. The user is also informed as to whether or not the program operates in an interactive environment. In many cases the user may need a main program and not merely a subroutine.

These requests are made to INFLIB by using program attributes and attribute values. For example, LANGUAGE, PRECISION and NAME are program attributes. The user may request a program written in FORTRAN. FORTRAN then becomes the value of the LANGUAGE attribute. A user may further define his request by giving the attribute LANGUAGE with the value FORTRAN, the PRECISION attribute with the value EXTENDED or DOUBLE, or the NAME attribute with the value ANYNAME.

INFLIB also allows a combination of values for every attribute. The user may request a program whose LANGUAGE attribute is FORTRAN or PL/I and whose MODE is either BATCH or INTERACTIVE.

INTERROGATING THE SYSTEM

INFLIB is a simple query language based on a number of keywords, which are attribute names. For example, PRECISION is the keyword representing the precision attribute. Keywords are followed by a logical expression enclosed in parentheses. The logical expression is a requested value or a combination of requested values for a particular attribute. A user interested in a program, RANDOM1, for example, should request it by typing:

```
NAME(RANDOM1);
```

Where NAME is the keyword or attribute and RANDOM1 is the logical expression or attribute value. If the user needs a program in double precision, he should type:

```
PRECISION(DOUBLE);
```

The INFLIB system provides a number of logical operators which may be used to combine values and, thereby, reduce the number of requests necessary.

These operators are given in TABLE 1.

Operator	INFLIB Representation
OR	,
AND	& (or a blank)
NOT	¬

Table 1: INFLIB Logical Operators

The examples below illustrate the use of these operators.

Example 1:

A user interested in programs in either library RANDOM1 or RANDOM2 would code:

NAME(RANDOM1,RANDOM2);

Example 2:

Assuming the user knows previously that RANDOM1 is not suitable, he may eliminate it by coding:

NAME(¬RANDOM1);

It is possible to combine a number of keyword attributes in one query as shown here.

1. TITLE(INTERPOLATION) LANGUAGE(FORTRAN);
2. ALGORITHM(AITKEN,NEWVILLE) LANGUAGE(¬APL& FORTRAN) MODE(BATCH);
3. TITLE(INVERSE&MATRIX) PRECISION(SINGLE);

The Keyword TITLE refers to a one-line description of the program. Each query may be five records (cards) and must end with a semi-colon(;). When operating under TSO the user must end the session with the command: END;.

INFLIB also accepts messages from the user. This is done by entering a line of 80 characters or less starting with an asterisk (*) and ending with a semi-colon(;).

TABLE 2 lists the INFLIB Keywords and their abbreviations.

Keyword	Abbreviation
NAME	NAM
TITLE	TIT

ALGORITHM	ALG
LANGUAGE	LANG
PRECISION	PREC
MODE(INTERACTIVE OR BATCH)	MODE
TYPE(SUBROUTINE,MAIN,PACKAGE OR CATALOGUED PROC)	TYP
COMPILER	COMP
LIBRARY	LIB
NUMBER	NUM
SUBJECT	SUB
APPLICATION	APP
PUT(TITLE OR	PUT(TIT
DESCRIPTION OR	DES
USAGE OR	USE
ALL)	ALL)

The first nine entries in Table 2 are self-explanatory, however, further clarification is needed of the others. This section provides the necessary explanation of the keywords.

1. NUMBER

Every program has been assigned a unique number, which serves as a direct key to it. Program numbers comprised of five digits and two letters were created because program names are not always unique. There are many cases where different libraries contain programs with identical names.

2. SUBJECT(Values)

It is often the subject which provides the most useful approach to a program. Therefore, for every program there is a list of SUBJECT keywords. By correctly combining these keywords, the user may clearly define his needs to the system. For example, the user needs a program which performs interpolation or approximation. He may code:

```
SUBJECT(INTERPOLATION,APPROXIMATION);
```

If a program is needed for matrix inversion, the following may be coded:

```
SUBJECT(INVERSION & MATRIX);
```

To find a program for solving differential or integral equations, the user may code:

```
SUBJECT(DIFFERENTIAL & EQUATIONS, INTEGRAL & EQUATIONS);
```

Failure to clearly define the subject may result in a large amount of unnecessary or unwanted output.

3. PUT

The language command PUT allows the user to specify the type of information he requires about each program satisfying the query. The user may request information on the TITLE (a one line description of the program), DESCRIPTION, USAGE or ALL of the three previous fields. The request is enclosed in parentheses and follows the word PUT.

PUT(TITLE) - This is the default option and is assum if PUT is not included in the query. It lists the program name, number and title.

PUT(DESCRIPTION) - This option lists further details on the program's purpose, language, precision, type, region size and basic operations.

PUT(USAGE) - This lists all the operating instructions, including parameter explanation, library location and documentation. It may be requested together with DESCRIPTION.

PUT(ALL) - This option lists all of the above information as well as information on the compilers, algorithms, author name, consultant and any other pertinent remarks.

SUMMARY OF INFLIB PRINCIPLES

In summary, it can be briefly stated that INFLIB is a query language which uses a set of keyword attributes as well as the command PUT to find a requested program and to control the information given in response. Every keyword is followed by a logical expression enclosed in parentheses. This expression may be a value or a combination of values required for a specific attribute. Expression values are combined by using the operators AND(&), OR(,) and NOT(¬). Each query into the system may refer to any number of attributes in any order and may be combined using the logical operators. As a matter of convenience, INFLIB

also allows a blank within the parentheses to replace the operator AND.

The user is given the option of adding information into the system in the form of a comment. This is done by entering a line of 80 characters or less which starts with an asterisk and ends with a semi-colon. Such lines are saved and later submitted to the Users' Group who will evaluate them and make changes to the system accordingly. It is highly recommended that users take advantage of the comment feature as it serves as feedback to the Users' Group and is the most effective method of keeping the system accurate and up-to-date.

INTERNAL ORGANIZATION

The INFLIB system is built on a main data base, two secondary datasets, and a dataset for storing feedback information. The main data base has indexed-sequential organization with program numbers as keys and contains all the recorded information available for a particular program. (Under OS/VS, the data base will have VSAM organization). The two secondary datasets serve as secondary indexes for subject keywords and program names. Every item in the main data base may be directly accessed by program number with the subject or name serving as alternative keys. This design makes it possible to efficiently retrieve responses. The secondary indexes, referred to as lexicons, also deal with the interrelationship of synonyms and related words. System operations are traced and stored on the third dataset, thereby providing a feedback feature for the system. With this information it is possible to constantly update the main data base.

The effectiveness of the entire system depends on the completeness of the main data base. It must contain all the available information on all the software known to the computing center in order to overcome the existing problem of diverged documentation. The details in the data base must be written clearly and exactly. It is therefore, recommended that the details be written by the program's author or the person responsible for the program's maintenance. These details should then be submitted to a system supervisor who makes the final decision on what is entered and its accurateness.

There are two main difficulties in the keyword selection. First, there are often a number of spellings for the same word or a number of words stemming from the same root. Second, there are many

words with more than one meaning and often a number of words define the same expression.

The system has overcome the first problem by including synonyms in the secondary lexicons which relate equivalent words to one another.

The second problem has been partially overcome by initiating a method of "teaching" the system new words. New keywords presented through system queries are recorded in the trace or feedback dataset described above and the system is updated accordingly.

One of the most important features of the system is its flexibility and expandability. This is provided by the trace or feedback dataset. The system records all queries and comments made to it. This provides the program designers with constant feedback and the possibility of making periodic additions and clarifications. The system may be easily expanded by adding commands and keywords to the query language or by adding new lexicons to the existing data.

קבוצת דיון מס׳ 12:

"איחזור ועיבוד מידע טקסטואלי"

יו"ר: ד"ר י. שויקה

ח. שור

ע. אורנן

"נירתוח מבחין" (Discriminant Analysis)

מירושם לטקסט מקראי

חיים שור את יהודה ת' רדאי

זה מכבר בתעוררו ספקות בדבר ההומוגניות של ספר שופטים. ספקות אלו שואבים תוקפם מהחריגות התוכנית של חטיבות הספר השונות, יחסית לגופו העיקרי של הספר הכולל את הפרקים ג' 7 - י"ב, להלן חטיבה ב'. החטיבות אשר בהן אובחנה חריגות תכנית הן:

- ההקדמה (פרקים א'1 - ג'6) להלן חטיבה א';
- סיפורי שמשון (פרקים י"ג - ט"ז), להלן חטיבות ג' וד';
- מעשה פסל מיכיהו (פרקים יז' - יח'); להלן חטיבה ה';
- מעשה פילגש בגבעה (פרקים י"ט - כ"א), להלן חטיבה ו';

לצורך בדיקת ההומוגניות של החטיבות השונות ושל ספר שופטים כולו כיחידה אחת נבחרו (על-סמך עבודות קודמות של רדאי את ויקמן), 25 קריטריונים בלשניים כמותיים. באמצעות קריטריונים אלה נערכו מבחנים סטטיסטיים, אשר אוששו את השערת-האפס, כי כל חטיבה היא הומוגנית בפני עצמה ונכתבה בידי מחבר אחד. בנוסף לכך התברר, כי אין יסוד סטטיסטי להפרדה בין החטיבות ג' ו-ד' (סיפורי שמשון) והוא הדין לגבי חטיבות ד' ו-ה' (מיכיהו והפילגש). ממצאי מחקר זה עומדים לצאת בקרוב לאור במאמר לועזי.

בעוד שהמבחנים הסטטיסטיים, בהם נעשה שימוש משיבים לשאלה, אם יש לדחות את השערת-האפס בדבר מוצא משותף אם לאו, הרי אין הם מאפשרים "מיפוי גיאוגרפי" של המרחקים בין החטיבות השונות, על סמך הדמיון בין ערכי הקריטריונים, ואף לא בדיקת מידת ההומוגניות שלהן. הטכניקה הסטטיסטית, בה נעשה שימוש במחקר הנוכחי, מאפשרת מדידה כמותית של מידת הדמיון בין החטיבות השונות וכן מאתרת את ההיבטים (או המימדים) השונים, לפיהם ניתן להבחין בין חטיבות, אשר מבחנים סטטיסטיים קודמים מצאו, כי הן הטרוגניות.

לבסוף, מאפשרת הטכניקה, שתוצג להלן, לבצע שקלול הולם של הקריטריונים השונים על פי עצמת ההבחנה, שהם מגלים בין חטיבות הספר.

טכניקה סטטיסטיקה זאת, הנקראת "בתוח מבחין", מטרתה בניית קריטריון רב-משתני למיון מקרים (או מדגמים) לאוכלוסיות ידועות, בהתאם לנתונים על משתנים שונים, שהוגדרו ביחס לאותם מקרים. לשם דוגמה, נניח שלרשותנו 50 טקסטים, שנכתבו בשפה העברית ו-50 טקסטים בשפה האנגלית, ונידון בתונים ביחס לכל אחד מהטקסטים על 20 משתנים בלשניים שונים, כגון: שכיחות המלים בנות שלוש הברות, שכיחות ה' היידוע וכיוצא באלה. אם נבדוק את הערכים של משתנים אלה בטקסטים (מדגמים) השונים, נמצא, כי קיימים הבדלים בין המשתנים במדה, בה הם מאפיינים את השפה, שבה נכתב הטקסט. למשל, ידוע, כי ביחס למשתנים המייצגים את אורך המלה ישנם הבדלים ניכרים בין טקסטים עבריים לטקסטיים אנגליים, מה שלא כן כנראה ביחס לשכיחות ה' היידוע.

בעזרת "הניתוח המבחין" נבנית משוואה ליניארית של המשתנים שנמדדו, ובאמצעותה יחושב "ציון מבחין" (Discriminant Score) לכל אחד מהטקסטים הנבדקים. מקדמי המשתנים במשוואה זאת, הנקראת "פונקצית האבחנה", נקבעים באופן כזה, שההבדלים ב"ציונים המבחינים" של טקסטים השייכים לאותה אוכלוסיה לטובית יהיו קטנים מאוד, בעוד שההבדלים בין "ציונים מבחינים" של טקסטים השייכים לאוכלוסיות שונות יהיו גדולים מאוד. במלים אחרות: ב"פונקצית האבחנה" משקל גדול (מקדמים גבוהים) למשתנים המאפיינים אוכלוסיה לטובית ומשקל אפסי למשתנים, שאינם מייחדים אוכלוסיה.

"פונקצית האבחנה", לאחר שנבנתה על סמך מדגמים, ששייכותם הלטובית ידועה, תאפשר לנו לסווג כל טקסט, אשר ערכי המשתנים הבלשניים שלו ידועים, אך אוכלוסית המוצא שלו אינה ידועה. על-ידי הצבת ערכי המשתנים ב"פונקצית האבחנה" נמצא, מהו ה"ציון המבחין" של אותו טקסט, וציון זה יסייע לסווגו לאוכלוסיה המתאימה תוך ציון ההסתברות, כי הסיווג שנבצע הוא נכון.

"פרבקצית האבחנה", שבאמצעותה מחושבים "הצירובים המבחינים" מתארת בדרך-כלל היבט או מימד אחד בלבד של האבחנה בין האוכלוסיות הנבדקות. כאשר קיימים היבטים אחדים כאלה וכאשר מספר אוכלוסיות המוצא גדול משתיים, נקבל כמה וכמה "פרבקציות אבחנה", אשר כל אחת מהן מייצגת היבט אבחנה שונה בין האוכלוסיות. הציון המבחין של טקסט מסויים בפרבקצית אבחנה בתרבה מהווה איפוא ציון של הטקסט ביחס למימד אחד של אבחנה. לדוגמה: במחקר, אשר ממצאיו יוצגו להלן, נמצא, כי ניתן להבחין בין חטיבות ספר שופטים על בסיס של שני גורמים בלתי-תלויים זה בזה; (א) התפלגות מספר ההברות במלה, (ב) התפלגות המעברים בין חלקי-הדיבר.

ואמנם, מעיון במקדמים של שתי "פרבקציות האבחנה" שהתקבלו עולה, כי בפרבקציה הראשונה ישנו משקל רב (מקדמים גבוהים) למשתנים המייצגים את מספר ההברות במלה (למשל תאחוז המלים בנות הברה אחת, בנות שתי הברות וכו'), ואילו בפרבקציה השניה יש משקל רב למשתנים המייצגים את התפלגות המעברים בין חלקי-הדיבר (למשל תאחוז המעברים משם-עצם לפועל, מפועל לשם-עצם וכיו"ב).

כיוון שלכל טקסט חושבו מספר "צירובים מבחינים" (ציון מבחין אחד לכל פובקצית אבחנה) וכיוון שצירובים אלה מתארים מימדים בלתי-תלויים של אבחנה בין הקבוצות, ניתן להציג כל טקסט כנקודה במערכת צירים מרחבית, אשר בה כל ציר מייצג פובקצית אבחנה שונה, כלומר: מימד אבחנה שונה.

אם נתרכז בשני מימדי האבחנה החשובים ביותר, אפשר יהיה לתאר את הטקסטים באופן גראפי במערכת צירים דו-מימדית. הצגה גראפית כזאת תאפשר לחקור את היחסים ההדדיים שבין הטקסטים השונים ולהגיע למסקנות באשר למרחקים הגיאומטריים (מידת השוני) שבין טקסטים שונים או בין אוכלוסיות מוצא שונות.

./..

במחקר המדורח כאן נעשה שימוש ב-25 קריטריונים המתחלקים לשלוש קבוצות כדלהלן:

קבוצה א': קריטריונים של אורך המלה.

V1 - תאחוז המלים בנות הברה אחת
V2 - " " שתי הברות
V3 - " " " שלוש "
V4 - " " " ארבע "
V5 - " " " חמש " ומעלה.

קבוצה ב': קריטריונים המייצגים מעברים בין חלקי הדיבר.

V8 - תאחוז המעברים משם-עצם לשם-עצם
V9 - " " " " לפועל
V10 - " " " " לחלק דיבר-אחר
V11 - " " " " לסוף-פסוק
V12 - " " " מפועל לשם-עצם
V13 - " " " " לפועל
V14 - " " " " לחלק דיבר-אחר
V15 - " " " " לסוף-פסוק
V16 - " " " מחלק-דיבר אחר לשם-עצם
V17 - " " " " " " לפועל
V18 - " " " " " " לחלק-דיבר אחר
V19 - " " " " " " לסוף-פסוק
V20 - " " " מסוף-פסוק לשם עצם

שור את רדאי

קבוצה ג': קריטריונים שונים

V6 – תאחוז הפעלים בעבר, בעתיד ובציווי (Finite Verbs) בכלל הפעלים
V7 – " הסבילים בכלל הפעלים
V21 – " שמות-העצם הנסמכים בכלל שמות-העצם
V22 – " " " הנפרדים " "
V23 – " מליות-הקישור בטקסט כולו
V24 – " ו' החיבור בטקסט כולו
V25 – " ה' היידוע במלים, בהן היא עשויה להופיע בכוח

כל אחת משש חטיבות הספר חולקה באקראי למדגמים שווי-גודל בקירוב כפי שמראה לוח מס' 1.

לוח מס' 1: חלוקת חטיבות ספר שופטים למדגמים

חטיבה	מס' מדגמים	מס' מלים למדגם	סה"כ מלים בחטיבה
א	5	192	959
ב	18	251	4512
ג	4	263	1051
ד	3	185	552
ה	4	184	734
ו	6	271	1625
סה"כ	40		

כל אחד מהמדגמים נותח באמצעות מחשב ונמצאו 25 ערכי המשתנים, שהוגדרו לעיל, ביחס לכל מדגם ומדגם.

./..

בשלב הראשון של הניתוח הסטטיסטי חולקו 40 המדגמים לשש קבוצות (שש אוכלוסיות מוצא), בהתאם לשש חטיבות הספר, תוך התעלמות גמורה ממימצאי המחקר הסטטיסטי הנזכר של רדאי את ויקמן, שגילה כאמור, כי אין יסוד להפרדה בין חטיבה ג' לד' מזה ובין חטיבה ה' לו' מזה.

באמצעות תכנית מורכבה השייכת לקובץ התכניות S.P.S.S בערך "ניתוח מבחין" ונתקבלו שתי "פונקציות אבחנה" באמצעותן חושב "ציון מבחין" כפול לכל מדגם. ציונים אלה מתוארים תיאור גראפי בדיאגרמה מס' 1. ממוצעי "הציונים המבחינים" בחטיבות השונות קובעים את "מרכזי הכובד" שלהן והם מסומנים בדיאגרמה בכוכבית.

המרחקים הגיאומטרים בין "מרכזי הכובד" של החטיבות נתונים בלוח מס' 2.

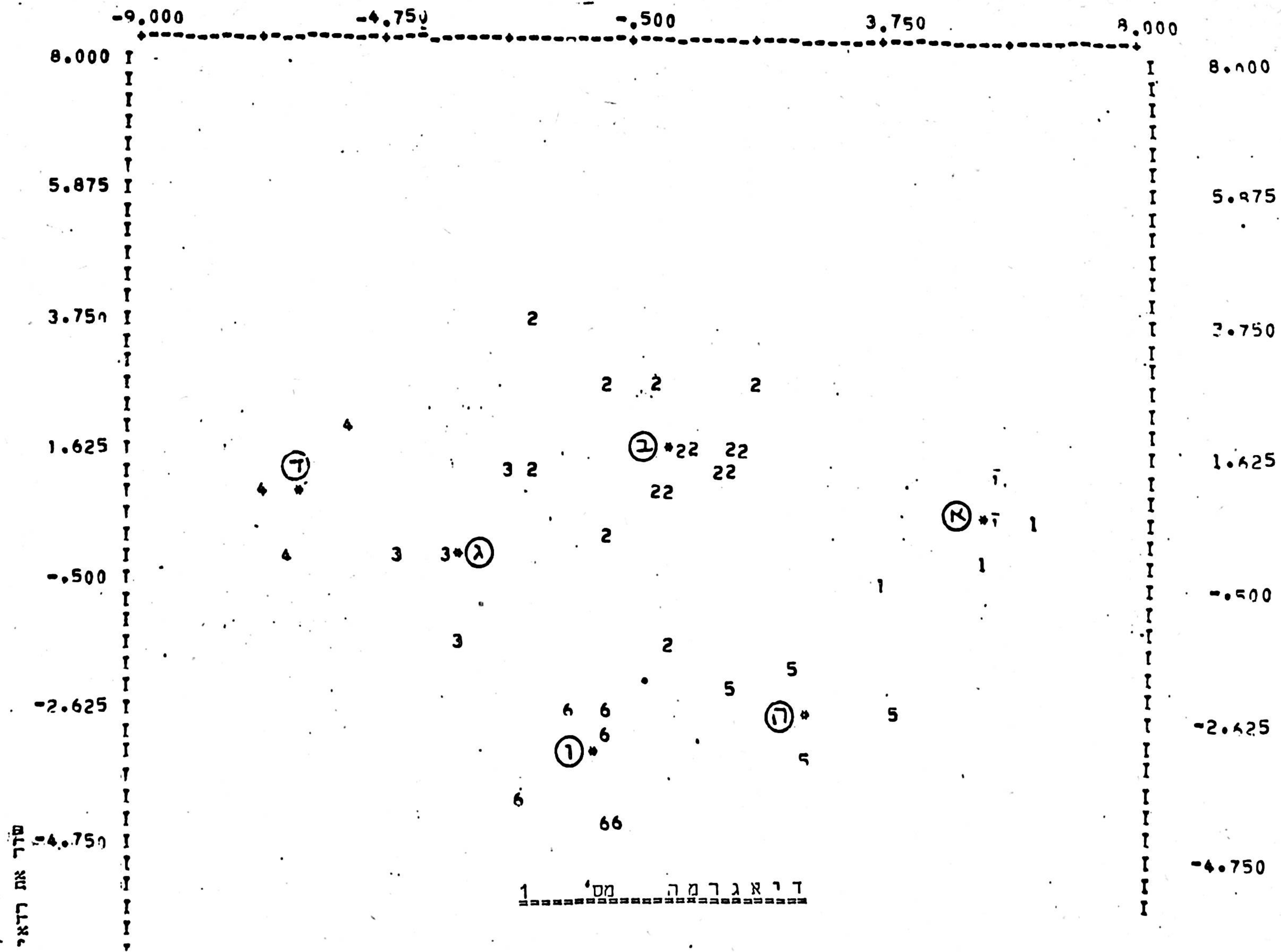
לוח מס' 2: המרחקים בין "מרכזי-הכובד" של החטיבות (חלוקה ל-6 חטיבות)

חטיבה	ב	ג	ד	ה	ו
א	6.5	10.7	14.0	5.2	9.1
ב		4.8	7.7	5.8	6.1
ג			3.5	7.7	4.7
ד				11.2	8.0
ה					4.4

ניכר בעליל, כי המרחקים הקצרים ביותר הם בין חטיבות ג' ו-ד' (3.5) ובין חטיבות ה' ו-ו' (4.4). הואיל וכך, ובהסתמך על מבחנים סטטיסטיים קודמים, יש מקום לאחד שני זוגות חטיבות אלה. 40 המדגמים חולקו איפוא מחדש לארבע חטיבות: חטיבה א', חטיבה ב', חטיבה (ג'+ד') וחטיבה (ה'+ו').

./..

JT OF DISCRIMINANT SCORE 1 (HORIZONTAL) VS. DISCRIMINANT SCORE 2 (VERTICAL). * INDICATES A GROUP CENTROID.



בהתאם לחלוקה חדשה זאת של מדגמי הספר לארבע אוכלוסיות מוצא שונות בוצעה הרצה נוספת של התכנית "ניתוח מבחין" והתקבל לוח המרחקים הבא (לוח מס' 3).

לוח מס' 3: מרחקים בין "מרכזי-הכובד" של החטיבות (חלוקה לארבע חטיבות)

חטיבה	ב	ג+ד	ה+ו
א	7.5	13.0	8.5
ב		6.0	6.8
ג+ד			8.0

נמצא, כי חטיבה א' מבודדת משאר החטיבות. תוצאה זאת תואמת בדיוק את הפרשנות הקוראית הרואה בחטיבה זאת גוף זר בספר שופטים (שמא מעבר מספר יהושע או השלמה לו, אשר נכתבה בידי עורך מאוחר, שמא כקאופמן, שריד של ספר שאבד). נראה מוצדק איפוא להתייחס לחטיבה א' כ"רעש" המפריע להבחנה בין החטיבות הנותרות ולאיתור היבטים המשמשים בסיס להבחנה. אי-לכך, הוצאה חטיבה א' מכלל ניתוח נוסף, ונערך ניתוח מבחין שלישי בשלוש החטיבות הנותרות (ב', ג'+ד', ה'+ו'). מהרצה זאת של תכנית "ניתוח מבחין" התקבלה דיאגרמה מס' 2.

בהשוואה לדיאגרמה הקודמת (מס' 1) ניכרת בבירור בדיאגרמה הזאת ההבחנה בין שלוש החטיבות העיקריות של ספר שופטים, ולא נמצאה כל חפיפה בין האזורים, בהם ממוקמים מדגמים השייכים לחטיבות שונות.

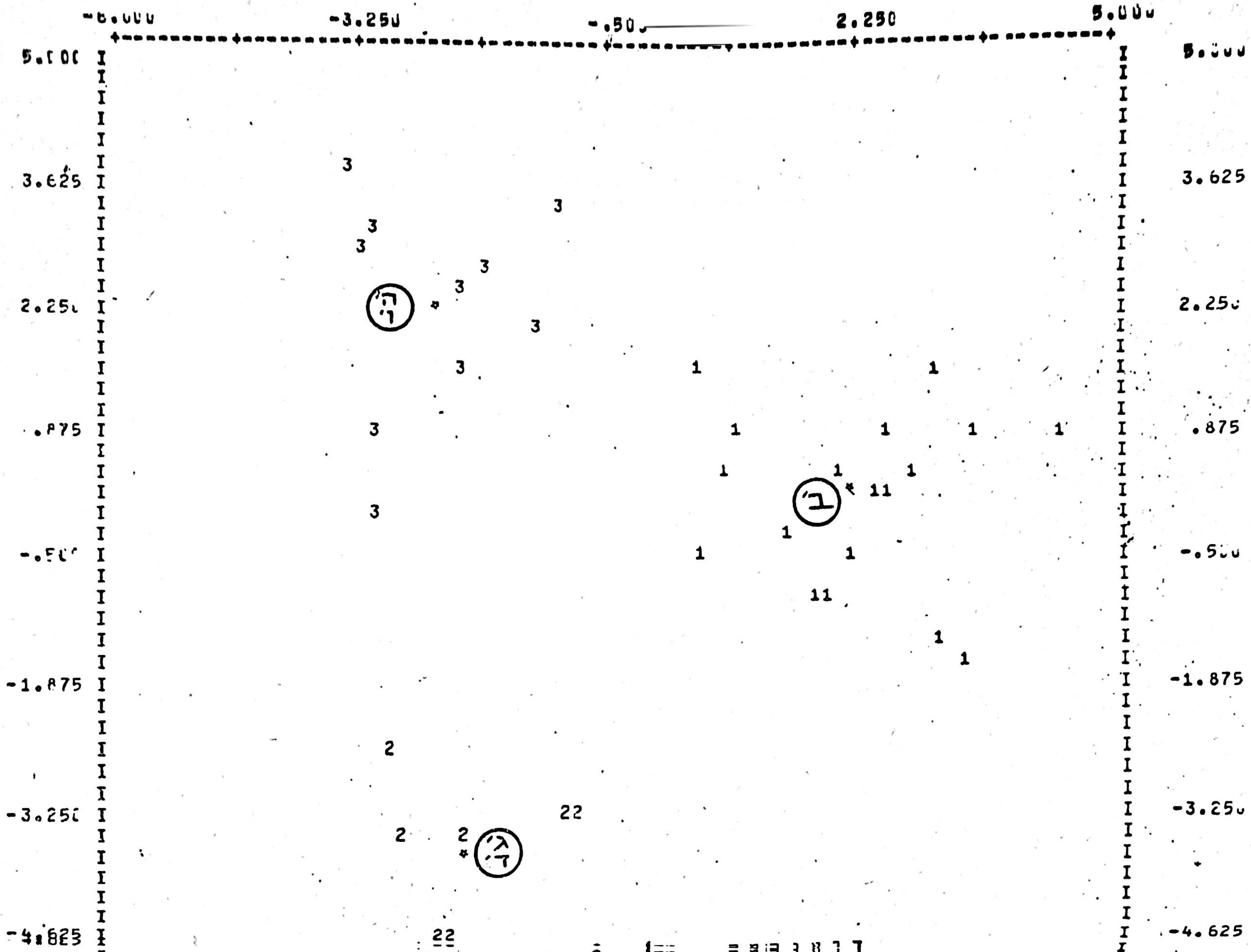
לוח מס' 4 מביא את מקדמי המשתנים המופיעים ב"פונקצית האבחנה" הראשונה (ציר x בדיאגרמה מס' 2) "בפונקצית האבחנה" השנייה (ציר y).

./..

לוח מס' 4: מקדמי המשתנים בפרבקצירת האבחנה

STANDARDIZED DISCRIMINANT FUNCTION COEFFICIENTS .

	1	Rank	2	Rank
V1	6.75236	(2)	1.45766	
V2	8.69540	(1)	-.95500	
V3	6.35344	(4)	-.56206	
V4	4.14767		1.83528	
V5	.09618		-.56968	
V6	1.21347		.81110	
V7	-1.82936		-2.02863	
V8	-6.67971	(3)	1.66193	
V9	-1.77790		-2.72562	(3)
V10	-1.66653		1.83320	
V11	-2.75981		1.60283	
V12	-4.52789		3.17506	(1)
V13	-2.04466		2.51055	(4)
V14	-2.96316		3.04924	(2)
V15	-2.86518		1.63955	
V16	-4.25477		1.39173	
V17	-1.19040		.78983	
V18	-5.56841		.79638	
V19	-3.99540		-.36246	
V20	-.42171		.84104	
V21	1.01939		-1.95293	
V22	-.80014		-2.34892	
V23	.40581		-2.28880	
V24	-2.21179		-2.49389	
V25	-.70811		-.21777	



ארבעת המקדמים הגבוהים ביותר (בערכם המוחלט) מסומנים לפי סדר הולך ויורד, בספרות המוקפות במעגל. המשתנים, שאלה מקדמיהם, הם בני המשקל הגבוה ביותר בקביעת "ציון ההבחנה". עיון איכותי בלוח הזה מגלה, כי בפונקציה הראשונה משקל מכריע למשתנים המייצגים את אורך המלה, בעוד שבשנייה מכריעים המשתנים המייצגים את המעברים בין חלקי-הדיבר. ממצא זה מאפשר לאפיין ולהגדיר את מימד האבחנה המיוצג על-ידי כל אחת "מפונקצית האבחנה".

אם נשוב עתה ונתבונן בדיאגרמה מס' 2, נמצא, כי מימד האבחנה הראשון (ציר x) המתבסס על אורך המלה מבחין בין גוף הספר (חטיבה ב') מזה, לבין סיפור שמשון וסיפורי מיכיהו והפילגש מזה. לעומת זה מבצע מימד האבחנה השני (ציר y) המתבסס על שכיחות המעברים בין חלקי-הדיבר אבחנה נוספת בין סיפורי שמשון מזה לסיפורי מיכיהו והפילגש מזה. שני מימדים אלו הנם בלתי-תלויים זה בזה ומבצעים אבחנה כפולה בין חטיבות ספר שופטים.

מדדים ומבחנים סטטיסטיים שונים (כגון קורלציה קנונית ומבחן כי-בריבוע) מבססים את האמינות הסטטיסטית של המסקנות שהוסקו.

לבסוף יצוין, כי מפאת חוסר מקום אין דו"ח זה שלם, ומובאים בו עיקרי הממצאים בלבד. דיון ממצה יותר בממצאי המחקר יתפרסם במקום אחר.

עוזי אורנן

אחזור ועיבוד מידע עברי באותיות לטיניות

א. שיטת העיבוד המקובלת של "טכסט מלא"

מאמצים רבים הושקעו אצלנו בפיתוח שיטות אחזור ועיבוד טכסטים לצורכי שליפת נושאים תוכניים ולשוניים, עריכת מלונים וקונקורדנציות ועוד, כגון תוכנית עיבוד השאלות ותשובות בשיטת "הטכסט המלא" הנערכת ע"י אוניברסיטת בר-אילן ומכון וייצמן, או תוכנית המלון האקדמי ללשון העברית, הנערכת ע"י האקדמיה ללשון העברית וכבר הביאה לעולם כמה קונקורדנציות כמוצרי לואי.

ככל הידוע לי מועתקים הטכסטים של השאלות ותשובות אל סרט מגנטי (או אל תקליט מגנטי) כמות שהם, אולי בעריכה קלה של הכתיב, ותוכנית רבת עוצמה מופעלת על הטכסטים האלה כדי לאפשר, בכל עת שמוגשת בקשה, לשלוף את המסמכים שצירוף מלים מסוים נמצא בהם. (צירוף זה יכול להיות, כמובן, לא רק של מלים השייכות במישרין לתוכן הנושא המבוקש, אלא גם של מלים המציינות את המקור המשפטי שדן בנושא המבוקש, ואשר מסתבר כי כל מי שהוסיף לדון בנושא ציטט אותו.[1])

למעשה, התוכנית פועלת לא על הטכסט עצמו, אלא על מלון או קונקורדנציה שכבר עובדו קודם לכן, בעת קליטת הטכסט, ובעצם אפשר לומר כי מלון זה וקונקורדנציה זו הם חלק ממלון או מקונקורדנציה מקיפים, הכוללים את כל הטכסטים שכבר נקלטו לתוכנית.

גישתם של האחראים לתוכנית זו היתה נועזה ביותר: הרי ידוע עד כמה רב משמעית היא כמעט כל מלה הכתובה בכתב העברי הבלתי מנוקד. כמו כן יודע כל המנסה לעיין בשאלה זו, כי חלק ניכר מפעולת הפענוח שמפענח הקורא כרוך בפסילות תחבירות של הצעת הקריאה, שאע"פ שזו מתיישבת יפה עם המורפולוגיה, היא נפסלת בגלל אי התאמות תחביריות, כלומר נפסלת על פי מידע הנשאב ממלים סמוכות למלה הנקראת, ולא מן המלה הנקראת עצמה.

למשל, במשפט כגון "חבצלת הפקידה הפקידה את כספה בבנק" נפסלת קריאת המלה השניה כהַפְּקִידָה לא מטעם צורני, שהרי המלה השלישית, השווה לה בדיוק בכתיבה, אכן כך תיקרא, אלא היא נפסלת משום שלא ייתכן שהמלה

---

1) י. שויקה, "אחזור פסקי דין בשיטת הטכסט המלא", __החוק והמשפט בעידן המחשב האלקטרוני__, פרסומי המכון לקרימינולוגיה, 18, י-ם תשלא, עמ' 40 ואילך.

השלישית ~~[illegible]~~ הַפְּקִידָה, שכן -תואר חייב להיות צמוד אל הגרעין-השם בלא חציצה, ואם אין אנו מוכנים לסבור שהיתה כאן חזרה על אותה מלה פעמיים, על כורחנו נאמר שהמלה השניה איננה הַפְּקִידָה, אלא הַפְּקִידָה. ולא הבאנו בחשבון בשיקולינו אלה כיצד מנע הידע התחבירי שלנו קריאות אחרות, שהן אפשריות מצד המורפולוגיה, כגון הִפְקִידָה, או הַפְקִידָה, שהתחביר לא יפסלן, למשל, במשפטים כגון "חבצלת - הִפקידה המנהל על כספי הבנק", או "חבצלת - הַפקידה על כספי הבנק, אין טובה ממנה!".

והנה בעלי התוכנה, ע~~ל~~כל פנים ככל הידוע לי עד כה, לא ניצלו כלל את חוקי התחביר כדי לפסול קריאות בלתי מסתברות, והסתפקו בתוכניות מורפולוגיות ומלוניות. אכן אפשר להבין ולהצדיק את גישתם זו: ההסתמכות על חוקי התחביר לצורך קביעת קריאה נכונה של צורה (שדמותה לרוב שונה מדמות הערך עצמו) כרוכה בקשיי קשיים, ולעתים החוק התחבירי הקובע יכול להיבחן רק על פי מלה הנמצאת במרחק רב מן המלה שאת קריאתה מבקשים לקבוע, ולעתים על פי מלים השייכות בכלל למשפט אחר.

במקום נסיון דל סיכויים לכבול את חוקי התחביר אל המחשב לצורך קריאה (מה שנעשה כנראה במוחו של האדם בעת הקריאה), העמיקו בעלי התוכנית אל המורפולוגיה, ולפי מה שנתפרסם, תוכניותיהם נראות גאוניות בתחבולותיהן ובדרכי פסילת קריאות לא מסתברות, דרכים העשויות בהשכל לפי מיטב החומר הדקדוקי הצורני שעמד לרשותם.[2]

ב. "החלוקה הדקדוקית"

עם כל החריפות שהושקעה בתוכניות אלה, יש לציין כי גם לאחר חלוקת המלה לחמשת מרכיביה - אות שימוש, קידומת, גרעין המלה, סיומת, כינוי - עדיין נצטרך לבדיקה ב"תוכנית מסובכת של קביעת הערך וחיפוש במלון, תהליך שהוא הרבה יותר איטי יחסית"[3]. גם אם מאז פרסום המאמר שציטוט זה לקוח ממנו נמצאו דרכים יעילות יותר ומהירות יותר - וזאת מבלי להביא בחשבון את הקידום הטכני של המחשבים, שנעשו מהירים יותר -

---

2) סבורני אמנם שתורת הצורות הקלאסית של הדקדוק העברי איננה יכולה לעמוד בתחרות עם תורת צורות אחרת, שקווים לה סרטט פרופ' בן-חיים באנציקלופדיה יודאיקה כרך 8, ושהרחבתיה במידת מה בסידרה של מאמרים, בייחוד על מערכת הפועל. ר' "נטיית הפועל כיצד?" בגליון החדש של כתב-העת "אורחות", היוצא בימים אלה, ושם גם ביבליוגרפיה על חיבורים קודמים.

3) מאיר שפירא ויעקב שַׁיְבֶּה, "ניתוח מיכאנוגראפי של המורפולוגיה העברית: אפשרויות והישגים", לשוננו כז-כח (תשכד), עמ' 362.

קשה להניח שהיה כאן שינוי מהירות מהותי, ובעיקרו של הדבר, רוב הניתוחים המורפולוגיים מצריכים עיון מרובה פי כמה ממה שנראה לעין.

למשל, ב"נספח 6" במאמרם הנ"ל מביאים המחברים דוגמה ל"חלוקה דקדוקית" – שהיא השלב הקודם ל"חלוקה האמיתית" המיוסדת על בדיקה במלון – של צירוף האותיות הבית, וניתנות שבע אפשרויות של "חלוקה דקדוקית" כדלקמן:

	אות שימוש	קידומת	גרעין	סיומת	כינוי
(1)			הבית		
(2)		ה	בית		
(3)	ה		בית		
(4)	הב		ית		
(5)			הבי	ת	
(6)		ה	בי	ת	
(7)	ה		בי	ת	

ואולם בדיקה בגופן של ההצעות הללו תראה כי משנבוא לבודקן במבחן המלון, נצטרך לערוך מספר הרבה יותר גדול של בדיקות מלוניות, שכן כל אחת משבע החלוקות עשויה להתפרש כבאת כוחן של יותר מאשר אפשרות אחת, כדלקמן:

	הקריאה הנבדקת	כדוגמת
חלוקה (1): גרעין.	הִבְיֵת	פִּרְסֵם
	הַבְיֵת	פַּרְסֵם
	הֻבְיַת	פֻּרְסַם
	הַבְיָת	אַרְגָּז
	הָבְיַת	אָסְבַּת
חלוקה (2): קידומת+גרעין.	הִבִּית	הִבִּיס
	הַבִּית	הַבִּיס
	הֵבִית	הֵכִין
	הָבִית	הָכִין
	הֻבְיַת	הֻבְטַח
חלוקה (3): אות שימוש+גרעין.	הֲבְיֵת?	הֲבְטֵל?
	הֲבֻיַּת?	הֲבֻטַּל?
	הֲבֵית?	הַחֵיק?
	הֲבַית?	הֲדַיג?
	הַבְיָת?	הַשְׁטָר?
	הַבֵּית	הַחֵיק

	הקריאה הנבדקת	כדוגמת
חלוקה (3) (המשך)	הַבַּיִת	הַדַּיִג
	הַבְּיָת	הַכְּפָר
	הֲבָיַת-?	הֲפָרַת(-הַבָּשָׁן?)
	הַבְיַת-?	הַשְׁבַת(-יְשָׁרִים?)
חלוקה (4): אות שימוש+אות שימוש+גרעין.		
	הַבְיָת?	הַבְיָד?
	הַבְיֵת?	הַבְעֵס?(סוֹפְרִים)
חלוקה (5): גרעין+סיומת.	הָבִיתָ	רָאִיתָ
	הָבִית	רָאִית
	הִבִּיתָ	צִוִּיתָ
	הִבִּית	צִוִּית
	הֻבֵּיתָ	צֻוֵּיתָ
	הֻבֵּית	צֻוֵּית
	הַבְיַת-	יַלְדַּת-
	הֶבְיַת-	עֶרְגַּת-(עֶרְגָּה)
	הָבְיַת-	חָכְמַת-
	הֶבְיַת-	חֶשְׁכַת-(חֲשֵׁכָה)
חלוקה (6): קידומת+גרעין+סיומת.	הִבֵּיתָ	הִטֵּיתָ
	הִבֵּית	הִטֵּית
	הֻבֵּיתָ	הֻטֵּיתָ
	הֻבֵּית	הֻטֵּית
	הֲבָיַת-	הֲפָרַת-(סֵדֶר)
חלוקה (7): אות שימוש+גרעין+סיומת.	הֲבָיַתָּ?	הֲגָמַלְתָּ?
	הֲבָיַתְּ?	הֲגָמַלְתְּ?
	הֲבִיַּתָּ?	הֲבִטַּלְתָּ?
	הֲבִיַּתְּ?	הֲבִטַּלְתְּ?
	הֲהֻבְיַתָּ?	הַהֻחְבַּרְתָּ?
	הֲהֻבְיַתְּ?	הַהֻחְבַּרְתְּ?

בסך הכל הובאו כאן למעלה מ-40 אפשרויות דקדוקיות של קריאה נכונה. אל יטעונו אפוא <u>שבע</u> "החלוקות הדקדוקיות" שמציע המחשב, שהרי החומר שהמחשב עוסק בו הוא הכתב הבלתי מנוקד, ורב-משמעותו של זה ניכרת לא רק במלים כשהן נתונות בתוך ההקשר, אלא גם במלים כפי שהן מתקבלות לאחר החלוקה הדקדוקית. ברור בכל אופן, שהמחשב יצטרך להציע לכל צורה נבדקת כמעט "תיקוני עיוותים" מצורת הערך שבמלון במספר רב של מקרים שה"ערך" שלהן הוא מדומה ואיננו נמצא כלל במלון,

ומספרם הסופי של הנסיונות לגלות ערכים אפשריים יעלה לא במעט על מספר הקריאות שפורטו כאן.

ג. קלט מנוקד

עבודה פחותה פי כמה נצטרך להשקיע, אם אל המלים שאנו מעתיקים אל הקלט נשכיל לצרף הנחיות ברורות באשר לקריאתה המכוונת של כל מלה ומלה בהקשר הנתון. עקרונית, בשיטה כזאת נוהגים בעלי תוכנית המלון האקדמי ללשון העברית. כל מלה מוסיפים לה עובדי המלון המלומדים את "סימני ההיכר" שלה, היינו, את ניקודה. ומובן מאליו שכנגד השקעה זו בכוח אדם מעולה, נחסכת כל העבודה הענקית של ניתוח מורפולוגי על ידי המחשב. יתר על כן, תוכנית המחשב כוללת "לימוד עצמי", היינו מלה שכבר הוסיפו לה המלומדים את ניקודה, כבר "נלמדה", וכשיגיע שוב צירוף האותיות הזה מן הטכסט לעיבוד במחשב, מייד יוסיף המחשב עצמו את הניקוד - או את הניקודים אם היו בעבר קריאות אחדות - כפי שלמד במקרים הקודמים. כפי שנמסר לי בעל פה, כ-80% או יותר מן הניקוד המיוחס למלים ע"י המחשב הוא ניקוד שהבודק את קריאת המחשב לאחר מעשה, מקבל אותו כניקוד הנכון. תוכנית המחשב של האקדמיה מכילה אפוא את מצאי הצורות האפשריות, ומייחסת כל אחת מהן, בעבודת אדם או בהנחיית אדם, אל הערך המתאים.

ממילא מובן שאם יבקש אדם תשובה לשאלה על צורה מסוימת, יוכל המחשב להשוות בלי קושי את הצורה המבוקשת אל המצאי החבוי בו ולהודיע על ממצאיו לשואל.

שיטת בר-אילן מעמיסה על המחשב עבודת ענק, ואולם אין היא מזקיקה עבודה רבה של בלשנים; שיטת האקדמיה מייתרת את עבודת הניתוח של המחשב, אולם יצירת הקלט מצריכה יגע רב של בלשנים ומומחים ללשון. נראה לי שניתן להציע שיטה אחרת, שתצריך הרבה פחות עמל-אדם והרבה פחות זמן-מחשב משמצריכות הן תוכנית בר-אילן והן תוכנית האקדמיה. הנני מבקש לתאר שיטה זו בסעיף הבא.

ד. הכנת קלט בכתב פונטי

המלה העברית, כמו כל המלים בשפת אנוש, מכילה הן עיצורים והן תנועות, וגם אם השאלה אם יש גבול ברור ביניהם עודנה משמשת נושא לויכוח, ברור ששתי קבוצות הגיים כאלה קיימות בכל שפה. התנועות בעברית אינן "סימני היכר" חיצוניים של מלה, שיש לה כביכול עמידה בפני עצמה גם בלא התנועות, אלא הן עצם מעצמה ובשר מבשרה של המלה. מפני סיבות היסטוריות שאינן כרוכות בשום תכונה מהותית של הלשון,

מושטש-בכתב העברי הרגיל חלק ניכר מן התנועות, והאותיות מסמנות בעיקר את העיצורים. זו הסיבה שמלים רבות הכתובות בשיטה זו של "חסכון" בסימון התנועות, נכתבות למעשה בדיוק באותו אופן, כי הגורמים המבדילים ביניהן אינם מסומנים, ונמצא שהמלים הכתובות הללו הן "רב-~~[illegible]~~ משמעיות".

לו היינו מסמנים בכתב גם את התנועות של המלה ולא רק את העיצורים שלה, מייד היתה נעלמת כמעט לחלוטין רב-המשמעות הטבועה בכתב העברי המקובל. מלבד זאת, מכפל העיצורים - שלא כמקובל בשיטות כתב אחרות - מסומן בכתב שלנו באות אחת בלבד, ולפיכך אי אפשר להבחין בין אות המסמנת עיצור יחיד ובין אות המסמנת מכפל עיצורים (המדובר במכפל פונימי, בלא קשר לשאלה אם מכפל זה מתבצע בדיבור). ע"י הויתור על סימון המכפל נוספת רב-משמעות אל הכתב.

והנה במדע הלשון מקובל זה עשרות שנים להשתמש באותיות לטיניות (ואחרות כתוספת) לשם רישום פונטי ופונמי של שפות שבלשן מבקש לתארן. אין כל חשיבות לשאלה אם השפה המתוארת היא שפה עניה באוצר מלים, מדוברת בלבד, בפי אנשי שבט נידח, או אם זוהי שפה עשירת מבעים, בעלת תרבות כתובה בת אלפי שנים; כזו גם זו אפשר להעלות אותן על הכתב באותם האמצעים המקובלים במדע הבלשנות. ושיטת כתב זו כוחה יפה, כמובן, גם כלפי הלשון העברית לאורך כל תקופותיה. מלים עבריות שיהיו כתובות כראוי בשיטה זו (למשל, לפי הפירוט שיובא בנספח) לא יהיו דו-משמעיות, אלא אם כן הן דו-משמעיות באמת, בשפה, ולא רק בכתב הגרוע. על כן, אם למחשב יוגשו טכסטים עבריים הכתובים בשיטה זו, מייד ייבטל הצורך בתוכניות הפנטסטיות, הארוכות והמסובכות ככל שהן מתוחכמות ומחוכמות, ודרכי אחזור ועיבוד טכסטים בעברית לא יהיו מסובכות יותר מאשר הדרכים המקובלות לאחזור ולעיבוד טכסטים של שפות אחרות, כגון צרפתית או גרמנית.

ייאמר כבר מעתה כי צרכנים שמסיבות שונות לא ירצו לקבל את הפלט באותיות לטיניות - ערכנו בשבילם תוכנית פשוטה בשפת פסקל, המקבלת כל טכסט באותיות לטיניות והופכת אותו לטכסט באותיות עבריות בכתיב המלא המקובל, תוך השמטת התנועות שאינן נכתבות, ביטול מכפלים ושינויים אחרים, שייכללו בדוגמאות.[4]

4) התוכנית נכתבה בגדר השיעור "בלשנות ומחשבים" שנתתי בתשלו לתלמידי המחלקה למחשבים ולתלמידי החוג ללשון העברית (במשותף) באוניברסיטה העברית, בעיקר ע"י שמואל פלג ויעקב טלינוביץ ובהשתתפות כל הכיתה.

כלומר, הצרכן יוכל לשאול את שאלותיו כהרגלו בכתב העברי המקובל, ואף את התשובות יוכל לקבל בכתב זה, כי אנשי השירות הם שיתעתקו את המלים הקובעות של שאלותיו לכתב הפונמי שהמחשב מבין, והתשובות עצמן יועברו בשלב האחרון אל התוכנית ההופכת אותן מן הכתב הפונמי באותיות לטיניות אל הכתב העברי הרגיל.

ראוי להתעכב רגע קט על תפקידו של "איש השירות", שלכאורה הוא עוסק אך בהעברה טכנית של מלים עבריות מן הכתב העברי המקובל אל הכתב הפונמי באותיות לטיניות. האמת היא שבהטלת תפקיד זה עליו אנו מנצלים את הכוח הכביר הטמון במוחו של האדם, כוח העולה פי כמה על התוכניות המתוחכמות ביותר שיש בידינו להפעיל במחשב, והעשוי לנתח ניתוח מהיר ומלא את הצורות הרב-משמעיות, ולקבוע את ערכה הפלוני של כל אחת מהן על חלקי התוספת שבה כהרף עין. כוחו של האדם, כאמור, מנצל לא רק את הידע המורפולוגי ואת הידע המלוני, אלא גם את הידע התחבירי, והוא עושה פעולות אלה, כנראה, לא בנפרד זו מזו, אלא כל הנחה מורפולוגית נבדקת מיד ולחוד ב"מלון" שבראשו, ונבחנת על פי קשריה התחביריים של הצורה בהקשרה. עדיין לא הגענו לחיקוי מושלם של עבודת מוח האנוש, גם לא, לחילופין, לתוכניות שוות ערך לה בדרכים אחרות, אולי. על כן, אם נוכל לנצל לצורכי העבודה המעשית שלנו את תוצאות פעולתו של המוח - רווח וחיסכון יעמדו לנו ממקום זה.(5)

מושג על החסכון העצום שיושג נוכל לקבל אם ניתן דעתנו שוב ל-43 הקריאות השונות של הבית שפורטו לעיל, לעומת דרך הכתיבה בכתב הפונמי באותיות לטיניות. לו בזה האחרון היינו בוחרים, לא היו לפנינו אלא מספר מועט של מלים משמעיות, וכל אחת מהן היתה חד משמעית לגמרי:

הבֵית?	H-BEIT? (6)	הבֻיַּת?	H-BUYYAT?
הבַיִת?	H-BAYIT?	הבִיַּתָּ?	H-BIYYATTA?
הַבֵּית	HA-BEIT	הבִיַּתְּ?	H-BIYYATT?
הַבַּיִת	HA-BAYIT	הבֻיַּתָּ?	H-BUYYATTA?
הבִיֵּת?	H-BIYYET?	הבֻיַּתְּ?	H-BUYYATT?

---

5) מצד אחר, אין להתעלם מן התועלת העיונית העצומה שצמחה ודאי, או עתידה לצמוח, מתוכניות כגון תוכנית בר-אילן לניתוח מורפולוגי של מלה עברית כתובה בכתיב החסר המקובל. הרי בעצם זהו נסיון לבנות מודל להבנת חלק מתהליכי הקריאה של אדם יודע-עברית בטכסט עברי לא-מנוקד! זהו חלק בלבד, כאמור, משום שלא טופל עדיין בחיבור.

6) כדי להבחין בין ה' הידוע ובין ה' השאלה, רשמנו את זה האחרון בלא תנועה אחריו.

מכאן שלו היתה מזדמנת אחת המלים הללו בטכסט העברי הפונמי, לא היינו נזקקים לא ל"חלוקה מלאכותית" (או "צורנית") ולא ל"חלוקה דקדוקית", וממילא לא היינו מגיעים כלל לעבודת פענוח עצומה של 43 צורות של "מלה אחת רב משמעית". מה שהיה על התוכנית לעשות הוא להפריד בין בסיס המלה ("הגרעין") ובין מה שנוסף עליו (בלא להבדיל בין "סיומת ל"כינוי"), לאחזר את השינויים שחלו בבסיס עקב הנטיה, ולרשום את הערך המלוני.

ה. הכנת המלון

במפתיע מתברר כי בשיטה זו לא היה גם כל צורך לבנות מראש מלון עברי כלשהו. לא "מלון גרעינים" ולא "מלון אבן-שושן" חדש. שהרי ניתוח דקדוקי ראוי היה מעמיד לנו הוא עצמו בסוף כל ניתוח את הערכים המלוניים, וממילא היינו מקבלים "מלון", כלומר, רשימת ערכי המלים המצויות בטכסט מסוים.

מכאן, שכל טכסט וטכסט שהיינו מוסרים לעיבוד היה מעמיד את מלונו שלו, וכלל לא היינו צריכים להטריח מלון מודרני לשם עריכת מקור להשוואה עם טכסטים עתיקים, כגון שאלות ותשובות מן המאות שעברו, אלא הטכסטים הללו היו באים ומעמידים מלון לעצמם. ממילא, אם היה בא אדם ושואל על מלים מודרניות, היתה התשובה - מלים אלה אינן מצויות בנו.

ו. הערות לסיום

נותרו עוד נקודות שראוי להבהירן, ומפני קוצר המצע אעשה זאת כאן במלים מועטות בלבד.

(1) שיטת הכתב המוצע היא פונמית בלבד, בסטיות מועטות אל הכתיב הפונטי (כגון, פתח גנוב הננו רושמים, אע"פ שאינו תנועה פונמית). הכתיב משקף אפוא את מבנה המלה, ואין בו כל התימרות לשקף מבטא כל שהוא. למשל, ב מסומנת כ-B, ואחת היא אם מבטאה [b] או [v]. הקורא יידע לבטא כאן כראוי לפי כללים פשוטים. ליתר בירור ראה את מאמרי על ב׳כ׳פ׳ במקראה לתורת ההגה, תשלג, בהוצאת אקדמון (בעריכתי), או באנגלית בקובץ ההרצאות של הקונגרס הבין-לאומי ה-11 לבלשנות, בולוניה, 1973, חלק שני, ובקיצור בהקדמה לספר נטיית הפועל בתיאזר חדש, שחיברתי עם לביאה פיורקו.

(2) אף על פי שתנועות "קטנות" ו"גדולות" אין להן ערך פונמי, ועל כן רשמנו, למשל, הן קמץ והן פתח כ-A, הנה כדי להפוך את כתבנו הפונמי לכתב עברי רגיל, היה עלינו להבחין בין צירה מלא ובין שאר תנועות E.

כך זכה הצירה המלא לסימן מיוחד "Ei" בלא שהיתה לזאת סיבה פונמית, אלא לצרכים אורתוגרפיים בלבד. הוא הדין במה שקרוי "סגול מלא".

3) "אותיות השימוש" הופרדו בקו מן המלה עצמה. רעיון ישן נושן שאפשר כמובן להגשימו גם בכתב העברי המקובל, אלא ש"לא ייעשה כן במקומנו". אף על פי כן ראוי להזכיר כי רעיון זה כבר מומש בדפוס בשנות העשרים, בהוצאת "הספר" בלונדון.

4) ולבסוף תשובה לשאלה ה"מעשית" מי יכין את הקלט. התשובה: כל כתבן יודע עברית ונבון יוכל ללמוד בלי כל קושי את כללי המעבר מן הכתב העברי המקובל אל הכתב הפונמי העברי באותיות לטיניות. ברור שכתבן זה צריך יהיה לדעת דקדוק עברי במידה מספקת כדי לקבוע היכן עליו להשתמש באיזה כלל מעבר. על כן גיוס כתבנים יוכל להיעשות בשתי דרכים: או שאדם שמקצועו כתבנות יואות להשתלם בדקדוק וללמוד את כללי המעבר, או שאדם היודע דקדוק עברי יואות ללמוד כתבנות. בין כך ובין כך, אין ספק שכל כתבן ייתקל במלים שיתקשה בזיהוין, בייחוד אם יצטרך להעתיק טכסט של תקופה קדומה. בכל מקרה כזה עליו להעתיק אות כנגד אות ולציין את המלה בציון מיוחד, כגון * בסופה. המחשב יודרך להעלות מלים אלו ברשימה נפרדת בצירוף להקשרן, ובלשנים מומחים יתנו דעתם למלים אלה בלבד. לאחר שהבלשן ירשום את התעתיק המסתבר (או אף יותר מרישום אחד בו במקום!), ישבץ המחשב את הצורה המתוקנת שוב בטכסט.

- - - - - - - - - - -

<u>נספח 1: כללי המעבר</u>

<u>רשימה והעיצורים</u>

א	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	ל	מ	נ	ס	ע	פ	צ	ק	ר	ש	ת
+,7	B	G	D	H	W	Z	X	1,6	Y	K	L	M	N	S	^,&	P	C	Q	R	$	T

<u>רשימת התנועות</u>

כל תנועות A (ֳ,ַ,ָ) נכתבות A
חוץ מהֲ הנכתב H

כל תנועות E (ֱ,ֶ,ֵ) נכתבות E

: אינו מסומן בשום מקרה
(שווא אינו פונמה)

חוץ מצירה מלא הנכתב Ei, וכמוהו "סגול מלא"
כל תנועות i (חיריק מלא וחסר) נכתבות i, תנועות O (ֹ,ׂ,ֳ) נכתבות O
ותנועות U (קבוץ ושורוק) נכתבות U

<u>אותיות הסימוש</u> נכתבות כשקו אלכסוני / (או -) מפריד בין האות (או צירוף האותיות) ובין המלה.

<u>מכפל</u> יש לסמן בשתי אותיות, חוץ ממכפל שאחר אות שימוש, המסומן באות אחת.

<u>מכפל בסוף מלה.</u> מלים שבהיווסף אליהן הברה ייכפל עיצורן האחרון ראוי לכתוב מלכתחילה במכפל בסופן, כגון גן =GANN, סב ≙ SOBB. (פרט זה חשוב בייחוד אם הטכסט נועד לניתוח דקדוקי.)

<u>מלה המסתיימת בקמץ,</u> בלא ה אחריו (רָאִיתָ, שָׁמַעְתָּ) ייכתב בסופה הסימן =.

<u>שגיאות.</u> תוכנית המעבר מכתב עברי פונמי באותיות לטיניות לכתב עברי מקובל סובלת שגיאות כתיב מסוימות, ואת רובן היא מתקנת ממילא (בייחוד באשר למכפלים ולסימוני "י" ו-"יי"). אולם הניתוח הדקדוקי מבוסס על ההנחה שהכתב הפונמי הוא ללא דופי, ועל כן לא מן הראוי לשגות בתעתיק

<u>נספח 2: דוגמאות של פלט המחשב</u>

LIS MOLA DBAR HALAKA.	לשמוע דבר הלכה.
*YIKKANES, YYALLED, YERA+E	ייכנס, יילד, ייראה
*YI$MOE, YIXYE, BAYIT, RAGLAYIM	ישמור, יחיה, בית, רגליים
*+AYOM, MEYUXAD, HEXYEINU	איום, מיוחד, החיינו
*CE E MLE+	צרה מלא
*HETTER	היתר
*XAYYA+ WE/LO TAW , XAYYA	חייט ולו תו , חיה
*PANUY , MCUYYAN	פנוי , מצוין
*HIZKIR , HICCIL	הזכיר , הציל
*MI/BEITO , BNIYA	מביתו , בנייה
*GIYYORET , DIYYUN , SIYYER	גיורת , דיון , סייר
*QANO , TO+MAR , GOY	קנה , פאמה , גוי

BA/KOL +ADAM MITQANNE+; XUC MI/BNO W/TALMIDO.

בכל אדם מתקנא, חוץ מבנו ותלמידו.

YI$MAU +OZNEIKA= MA SE/PIKA= +OMER.

ישמעו אוזניך מה שפיך אומר.

אינדקס מחברים לפי א"ב

PROCEEDINGS

OF

THE NATIONAL CONFERENCE ON DATA PROCESSING

HAIFA 1976



INFORMATION PROCESSING ASSOCIATION OF ISRAEL



PROCEEDINGS

OF

THE NATIONAL CONFERENCE ON DATA PROCESSING

HAIFA 1976



INFORMATION PROCESSING ASSOCIATION OF ISRAEL